

From: Thomas, D., 2002, *Hacker Culture*, University of Minnesota Press, Minneapolis.

Chapter 1

Hacking Culture

Fry Guy watched the computer screen as the cursor blinked. Beside him a small electronic box chattered through a call routine, the numbers clicking audibly as each of the eleven digits of the phone number was dialed. Then the box made a shrill, electronic whistle, which meant the call had gone through; Fry Guy's computer . . . had just broken into one of the most secure computer systems in the United States, one which held the credit histories of millions of American citizens.

— Paul Mungo and Bryan Clough, *Approaching Zero*

This is the common perception of today's hacker — a wily computer criminal calling up a bank or credit card company and utilizing mysterious tools to penetrate simply and effortlessly the secure system networks that hold our most important secrets. However, any attempt to understand today's hackers or hacking that only examines the blinking cursors and whistling boxes of computing is destined to fail. The reason is simple: hacking has never been just a *technical* activity. Perhaps the most striking illustration of this is that William Gibson, who in his book *Neuromancer* coined the term “cyberspace” and who invented a world in which hackers feel at home, for nearly a decade refused to have an e-mail address. In fact, *Neuromancer*, the book that has been (rightly or wrongly) held accountable for birth of the “new breed” of hackers, is rumored to have been written on a manual typewriter.¹ Hacking, as Gibson's work demonstrates, is more about the imagination, the creative uses of technology, and our ability to comment on culture than about any tool, computer, or mechanism. The hacker imagination, like the literature that it is akin to, is rooted in something much deeper than microchips, phone lines, and keyboards.

The current image of the hacker blends high-tech wizardry and criminality. Seen as the source of many evils of high-tech computing,

from computer espionage and breaking and entering to the creation of computer viruses, hackers have been portrayed as the dangerous other of the computer revolution. Portrayals in the media have done little to contradict that image, often reducing hackers to lonely, malicious criminals who delight in destroying sensitive computer data or causing nationwide system crashes.

In both the media and the popular imagination, hackers are often framed as criminals. As Mungo and Clough describe them, hackers are members of an “underworld” who “prowl through computer systems looking for information, data, links to other webs and credit card numbers.” Moreover, hackers, they argue, can be “vindictive,” creating viruses, for instance, that “serve no useful purpose: they simply cripple computer systems and destroy data. . . . In a very short time it has become a major threat to the technology-dependent societies of the Western industrial world.”²

At least part of the reason for this impression rests with the media sensations caused by a few select cases. Clifford Stoll, as documented in his book *The Cuckoo's Egg*, for example, did trace a European hacker through the University of California at Berkeley's computer systems, ultimately revealing an attempt at international espionage, and the cases of Kevin Mitnick and Kevin Lee Poulsen, two hackers who were both arrested, prosecuted, and sent to prison for their hacking, gained considerable attention in the media and in subsequent books published about their exploits. Most of these accounts are journalistic in style and content and are more concerned with describing the events that took place than with analyzing the broader context out of which hackers have emerged. While the image of the hacker as a “criminal” seems to have taken over in the popular imagination, the broader context of the computer “underground” and, most important, the historical context force us to question such an easy categorization of this complex and varied subculture.

Between Technology and the Technical: Hacking as a Cultural Phenomenon

In the 1980s and 1990s, hackers were the subject of numerous films, TV shows, and news reports, most of which focused on the con-

nection between hacking and criminality. As Joe Chidley describes them in an article for *Maclean's Magazine*, "Hackers are people who simply love playing with computers," but there is a "malicious subset of the hacker community, who intrude on computer networks to do damage, commit fraud, or steal data," and these hackers "now have an arsenal of technologies to help them in their quest for secrets."³

Within such a limited framework, which reduces hackers to criminals with their "arsenal of technologies," it makes little sense to speak of a "culture" of hacking. Hacking appears to be, like most crime, something that malicious people do for reasons that don't always seem to make sense. Why would a talented computer programmer choose to write a virus rather than write a program that might be more useful and, potentially, economically more rewarding? Why would hackers break into unknown systems when their talents could be employed in many other more productive ways? These questions make the hacker's goals and motivations difficult to decipher.

Rather than attempting to understand the motivation behind hacking, the media and computer industry instead focus on the manner in which computers are hacked. At this level, hackers are easy to understand — they have a specialized set of tools, and they use those tools to commit crimes. This basic theme was central to the protest against the release of SATAN (Security Administrator Tool for Analyzing Networks), a network-analysis tool that tests systems for security flaws. The program, which was written to make system administrators aware of security flaws that were already well known and often exploited by hackers, was publicly released by its authors, Dan Farmer and Wieste Venema, in April 1995. The release was met with an outpouring of anxiety about the future of Net security and fear that the public availability of the tool would turn average computer users into criminals. As the *Los Angeles Times* remarked, "SATAN is like a gun, and this is like handing a gun to a 12-year-old."⁴ Other newspapers followed suit, similarly invoking metaphors of increasing firepower — "It's like randomly mailing automatic rifles to 5,000 addresses. I hope some crazy teen doesn't get a hold of one," wrote the *Oakland Tribune*, only to be outdone by the *San Jose Mercury News's* characterization: "It's like distributing

high-powered rocket launchers throughout the world, free of charge, available at your local library or school, and inviting people to try them out by shooting at somebody.”⁵ The computer industry was more sober in its analysis. “The real dangers of SATAN,” as one advisory argued, “arise from its ease of use — an automated tool makes it very easy to probe around on the network.”⁶

The basic objection to the release of SATAN was that it provided a tool that made system intrusion too easy, and making the program publicly available prompted outcries from those afraid that anyone with the tools could (and would) now invade systems. Omitted from most stories was the fact that SATAN had been available in a less powerful form as freeware (a freely distributed software package, accessible to anyone) on the Internet for years, along with several other programs that provided similar functions, not to mention a host of more powerful programs that were already widely available for the express purpose of unauthorized system entry. Additionally, SATAN only tested computer systems for well-known, previously discovered (and easily fixed) security holes.

SATAN was nothing new, but the discussion of it was. This response illustrated how convinced the general public was that the threat of hacking rested in the tools. While the apocalyptic effects of SATAN’s release failed to materialize (no significant increase in any system intrusion has been reported, nor has any been attributed to SATAN since its release), the anxieties that SATAN tapped into are still present. The response to SATAN was in actuality a response to something deeper. It was a reaction to a basic cultural anxiety about both the complexity of technology and the contemporary culture’s reliance upon that technology. SATAN appears to give anyone who wants it the tools to disrupt the system that very few people understand yet that everyone has come to rely on in their daily lives.

A cursory examination of both public and state responses reveals a paranoia regarding the hacker that one can easily attribute to a Luddite mentality, a generation gap, or a pure and simple technophobia that seems to pervade U.S. culture. While these aspects are a very real part of contemporary culture, such a simple set of answers covers over more than it reveals. Most of the responses to hackers and hacking have served to lower the level of public discussion by con-

fusing hackers with the tools that they use and making hyperbolic equations between computer software and high-power munitions. Like any other social and cultural phenomenon, the reasons for the growth of hacking in the United States (and as an international phenomenon) are myriad, and the reactions to hacking often reflect a wide range of reactions, from hope and fear to humor and dismay.

The responses to hacking — in the popular imagination and in the minds of agents of law enforcement and the criminal justice system, a response documented in court records, TV shows, movies, newspapers, books, and even Web pages — reveal more about contemporary culture than about hackers and hacking. However, much as was the case with SATAN, public reaction to hackers both tells us a great deal about the public that is reacting and, ironically, shields us from an understanding of the complexities and subtleties of the culture of the computer underground. By simply equating hackers with the tools they use, the media and popular representations of hackers have failed to understand or account for even the most basic motivations that pervade hacker culture.

In trying to determine what hacking is and what hacker culture looks like, I make a distinction between technology, as a broad, relational, and cultural phenomenon, and the technical or scientific, the products of technology itself (for example, telephones, computers, and modems).⁷ In doing so, I am also separating hackers' culture and motivation, which are very much about technology, from the idea of tools or specific technical items, which are for the most part incidental to the idea of hacking. These two concepts, technology and the technical, are different in kind, and to understand what constitutes hacking, we need to be careful to examine these two ideas as separate entities. Technology should be considered a *cultural* phenomenon, and in that sense, it tells us primarily about human relationships and the manner in which those relationships are mediated. The technical, by contrast, is concerned only with the instrumental means by which those relationships occur. It makes sense to speak of the technology of the telephone allowing people to have long-distance relationships. It also makes sense to discuss the technical aspects of telephones in comparison to the postal system. Both the phone and the mail as technology mediate human relationships in the same way insofar as

they allow us to communicate at great distances. Yet as technical phenomena they are completely distinct. To pose questions with respect to technology is to pose cultural and relational questions. To pose questions with respect to the technical is to pose instrumental questions. Put differently, to answer the question, What is hacking? properly, we cannot simply examine the manner in which hacking is done, the tools used, or the strategies that hackers deploy — the instrumental forces that constitute hacking. Instead we must look at the cultural and relational forces that define the context in which hacking takes place.

Hackers and Hacking

Not long ago, being called a hacker meant only that one belonged to a group of technology-obsessed college or graduate students who worked tirelessly on the dual diversions of finding interesting ways around problems (often in ways that resembled Rube Goldberg machines) and perpetuating clever, but harmless, pranks. This “class” of technophile is characterized by a kind of “moral code,” as documented by Steven Levy in his 1984 book, *Hackers*. The code, as Levy describes it, was “a philosophy, an ethic, and a dream,” and it was constituted by six basic theses:

1. Access to computers — and anything which might teach you something about the way the world works — should be unlimited and total. Always yield to the Hands-On Imperative!
2. All information should be free.
3. Mistrust Authority — Promote Decentralization.
4. Hackers should be judged by their hacking, not bogus criteria such as degrees, age, race, or position.
5. You can create art and beauty on a computer.
6. Computers can change your life for the better.⁸

The hackers Levy refers to were the original champions of the information superhighway, and their ethic was utopian in nature. As Levy describes it: “To a hacker a closed door is an insult, and a locked

door is an outrage. Just as information should be clearly and elegantly transported within the computer, and just as software should be freely disseminated, hackers believed people should be allowed access to files or tools which might promote the hacker quest to find out and improve the way the world works.”⁹

The “old hacker” of the 1960s and 1970s is often characterized with no small amount of nostalgia and is frequently seen as a counterpoint to the emergence of the new breed of hacker, the “cyberpunk” or “cracker.” The “old hackers,” in this romanticized telling, were “a certain breed of programmers who launched the ‘computer revolution,’ but just can’t seem to be found around anymore. . . . [A]ccording to these ‘old-school’ hackers, hacking meant a willingness to make technology accessible and open, a certain ‘love affair’ with the computer which meant they would ‘rather code than sleep.’ It meant a desire to create beauty with computers, to liberate information, to decentralize access to communication.”¹⁰ In short, the old-school hacker was dedicated to removing the threat of high technology from the world by making that technology accessible, open, free, and “beautiful.” To the 1960s hacker, hacking meant rendering technology benign, and hackers themselves not only were considered harmless but were framed as guardians of technology — scientists with an ethic that resembled Isaac Asimov’s “Laws of Robotics”: above all else, technology may never be used to harm human beings. Moreover, these hackers effected a strange anthropomorphism — information began to be personified, given a sense of Being usually reserved for life-forms. The old-school hacker was frequently motivated by the motto “Information wants to be free,” a credo that attributed both a will and an awareness to the information itself. Interestingly, it is these two things, will and awareness, that seem to be most threatened by the evolution of technology. In an era when the public is concerned both with a loss of freedom to technology and with a fear of consistently finding themselves out of touch with the latest technological developments, there is a transference of our greatest fears about technology onto the idea of information. The hacker ethic remedies these concerns through the liberation of information. The logic is this: if technology cannot even confine information, how will it ever be able to confine us? Within

the framework of this initial question we can begin to trace out the history of hacking as a history of technology.

A Genealogy of Secrecy

One of the primary issues that hackers and hacker culture negotiates is the concept of secrecy that has evolved significantly and rapidly since World War II. Indeed, hackers' relationships to technology can be understood as a cultural phenomenon and cultural response to the evolution of secrecy, particularly in relation to the broader political and social climate, the birth, growth, and institutionalization of the computer industry, and the increasing import of multinationalism in industry. The concept of secrecy seems to change from generation to generation. What secrecy means and particularly its value shift as social, political, and economic contexts change over time, but what has always remained stable within hacker culture is the need to negotiate the relationship between the technical aspects of the machines themselves and the cultural value of secrecy.

One of the first connections between secrecy and machines arose during the Allies' work to break German codes during World War II. Until this point, most cryptography had utilized methods of simple substitution, meaning that letters in the alphabet would be substituted for other letters, scrambling a clear text message into a ciphertext. For example, substituting the letter *a* for the letter *r*, *b* for *e*, and *c* for *d* would produce the ciphertext "abc" for the word "red." The problem with such a system of simple substitution is that the English language tends to utilize letters with fairly regular frequencies, and for that reason, no matter what substitutions are made, it becomes fairly easy to guess what has been encoded just by knowing how often certain letters appear in the message. Machines that encoded or decoded substitution schemes only helped speed up the process of encoding and decoding; they didn't actually perform the act of encoding in a meaningful way. That would all change with the German Enigma Machine, the first machine that actively encoded messages. The Enigma Machine consisted of eight code wheels that would rotate each time they were used. That meant that each time a substitution was made, the wheel making that substitution would

rotate forward so the next time that letter was to be used, the substitution would be different. To complicate things further, each of the eight wheels had a different set of substitutions. The only way to decode the message was to have an Enigma Machine on the other end, with the wheels set to the original position, and feed the message back through to decode it.

The process of World War II code-breaking spawned the first generation of computer scientists and committed the evolution of the machine to the interests of secrecy and national security. From World War II on, two of the primary functions of the computer would be code-making and code-breaking. Indeed, it is not too far a stretch to claim that the first computer scientists (including Alan Turing, the “father” of modern computer science) who broke the Enigma Machine’s coding process were, in the most basic sense, the first computer hackers and that the origin of computers rests with the need to keep and, perhaps more important, break secrets. The breaking of the Enigma codes would lead to the development of Colossus, what the museum at Bletchley Park (the site in Britain where the Enigma cipher was broken) calls “the world’s first computer.”

Secrecy has always been a significant concern for the military. Ever since Caesar scrambled messages sent to his troops, the military has recognized the need for secrecy. In the late 1930s and early 1940s, the stakes of military secrets were raised enormously with the invention and, ultimately, the use of the first atomic weapons. In that climate, the Department of Defense in the late 1950s turned to universities as the means to advance the study of computer science and engineering and, in turn, spawned what is generally acknowledged as the first generation of computer hackers. Funded almost exclusively by the Department of Defense, hacking began its difficult and oftentimes contradictory relationship to secrecy. As opposed to their forerunners, who worked exclusively in secrecy as they broke codes at Bletchley Park, hackers of the 1950s and 1960s, who worked in an environment of learning and academic freedom on university campuses, abhorred the notion of secrecy. And it was, in large part, that distaste for secrecy that led to most of the major advances that hackers would later make in the computer labs of MIT and other universities. “A free exchange of information,” Levy writes, “partic-

ularly when the information was in the form of a computer program, allowed for greater overall creativity.”¹¹

Such an ethic led to cooperation among programmers and a nearly constant evolution of ideas and programming techniques. The goal was not just to write original programs but to improve on the work of others. The most skillful hack was not writing a new line of code but finding a way to do something in someone else’s code that no one had seen before — to eliminate code, making the program run faster and more elegantly. While the ethic belonged to the hackers, the product belonged to the Department of Defense. That conflict, which most hackers of the 1950s and 1960s were either sheltered from or in denial about, represented the ultimate irony — work produced in the climate of absolute freedom would be deployed by the military in absolute secrecy. One hacker described the reaction to the realization that the funding and the results of those funded projects were military: “I got so upset I started crying. . . . Because these people had stolen my profession. They had made it impossible to be a computer person. They sold out. They sold out to the military uses, the evil uses, of the technology. They were a wholly owned subsidiary of the Department of Defense.”¹² Although these hackers were the “heroes of the computer revolution,” as Levy argues, they were also the brain trust that developed most of the hardware and software that would drive the military-industrial complex throughout the late twentieth century and create innovations that would be deployed in the name of secrecy and surveillance.¹³ The climate that hackers envisioned as “the realization of the hacker dream with sophisticated machines, shielded from the bureaucratic lunacy of the outside world,” was only realized up to a point. All the late-night hacks, high jinks, and clever pranks that characterized the 1960s hackers were furthering another agenda, which would have wider cultural impact than the hackers working on those projects imagined. An example that Levy cites is the funding of speech recognition by the ARPA (the Department of Defense’s Advanced Research Project Agency), a project that “directly increased the government’s ability to mass-monitor phone conversations at home and abroad.”¹⁴ Perhaps most significant was ARPA’s funding of ARPAnet, the foundation of the modern-day Internet, which was originally designed as a decen-

tralized communication network intended to maintain command, control, and communication abilities in the event of nuclear war. In essence, these early hackers designed a secure communications system funded by the Department of Defense to ensure survivability in nuclear war.

To the hackers of the 1960s, secrecy meant the freedom to share code in the computer lab, the spirit of cooperation in program design, and the right to tinker with anything and everything based on one's ability to improve upon it. For instance, "In a perfect hacker world," Levy writes, "anyone pissed off enough to open up a control box near a traffic light and take it apart to make it work better should be perfectly welcome to make the attempt."¹⁵ Freedom and secrecy were decontextualized to the point of solipsism. The hacker ethic, a matter of secrecy and freedom, was confined to the labs where hackers spent most of their waking hours and was naive about or ignorant of the greater context that was allowing that ethic to flourish. The shift that would make that naïveté most apparent was the move from the university to the corporate world. The second generation of hackers, the hackers of the 1980s, viewed secrecy differently. Without military funding and without corporate secrets to protect, these hackers took up the spirit of the original hacker ethic, but from a point of view that fully contextualized it in terms of politics, economics, and cultural attitudes. These hackers held to the tenets of freedom and abhorred the notion of secrecy, just as their predecessors did, but they lacked the solipsistic environment of the 1960s computer lab or the nearly unlimited government funding that made the original hacker ethic not only possible but also relatively risk-free.

Yesterday's Hackers

These original hackers of the 1950s and 1960s are generally recognized as the ancestors of the modern computer underground. There was, however, a second strain of hacker, one that is much more closely allied with the tradition of contemporary hacking. As Bruce Sterling writes, "the genuine roots of the modern hacker underground probably can be traced most successfully to a now

much-observed hippie anarchist movement known as the Yippies.”¹⁶ The modern underground, then, had its roots in a leftist political agenda that grew out of 1960s counterculture and was fueled by the antiwar protest movements of the 1960s and 1970s. “The Yippies, who took their name from the largely fictional ‘Youth International Party’ [YIP], carried out a loud and lively policy of surrealistic subversion and outrageous political mischief. Their basic tenets were flagrant sexual promiscuity, open and copious drug use, the political overthrow of any powermonger over thirty years of age, and an immediate end to the war in Vietnam, by any means necessary.”¹⁷ Jerry Rubin and Abbie Hoffman were two of the most high-profile members of YIP, but it would be two hackers, “Al Bell” and “Tom Edison,” who would take over the newsletter (initially called *Party Line*) and transform it into *TAP* (*Technical Assistance Program*). The vision of *TAP* became increasingly less political as it began to focus increasingly on the technical aspects of telephony. Ultimately, *TAP*’s primary mission became the distribution of information, for example, “tips on such topics as lock picking, the manipulation of vending machines, do-it-yourself payphone slugs and free electricity.”¹⁸ Although the war in Vietnam had served as the origin for *TAP*, by the time the war had ended, *TAP* had become a technical journal completely divorced from politics.

In the late 1960s and early 1970s, while the hackers of *TAP* were busily disseminating information, hackers at MIT were busy creating information. The majority of funding for most, if not all, of these hackers’ projects was coming from ARPA. The entire purpose of ARPA, begun in the early 1960s, was to create military applications for computers and computer networks. Although most of the hackers working at MIT either were blind to such considerations or found them irrelevant, hackers of the 1960s at major research universities were a significant portion of the technological side of the military-industrial complex (MIC). Hacker involvement in government projects with military application did not escape the attention of students protesting the war in Vietnam, and as a result computer labs at some major universities were the site of major protests and wound up being shielded by steel plating and half-inch-thick bulletproof Plexiglas.¹⁹ The role of the MIC was central,

although unintentional and unanticipated, to the formation of what constitutes the computer underground. On the one hand, the MIC produced an underground coming out of the protest movement of the 1960s. On the other hand, the MIC was funding the projects that would create hacker culture. Partially out of naïveté and partially because of the novelty of computers, the hackers of the 1960s and 1970s were able to avoid the obvious contradiction between their highly antiauthoritarian mind-set (“Information wants to be free”) and the fact that the people they were designing systems and software for were not likely to respect that basic tenet. Because they were producing the vast majority of new technology, the old-school hackers were able to maintain the illusion that they were also controlling it. Within a decade, the “old school” had moved to the Silicon Valley and started to build an industry that would look and operate increasingly less like the labs at MIT and Harvard and more like the corporations and organizations against which the 1960s hackers had rebelled.

Hacking the Industry: From the Altair to the Apple

Bill Gates: Cocky wizard, Harvard dropout who wrote Altair BASIC, and complained when hackers copied it.

Steven Jobs: Visionary, bearded, non-hacking youngster who took Wozniak’s Apple II, made lots of deals, and formed a company that would make a billion dollars.

Steven “Woz” Wozniak: Openhearted, technologically daring hardware hacker from San Jose suburbs, Woz built the Apple Computer for the pleasure of himself and friends.

— Steven Levy, “Who’s Who,” from *Hackers*

It may seem odd to think about hacking and hacker culture in relation to three of the most important figures from the personal computer (PC) industry. The computer industry has always been and in many ways continues to be the very antithesis of hacker culture, which is also the reason that it plays an important role in the forma-

tion of hacker culture. Without the hackers of the 1960s, there never would have been a PC; without the PC, there never would have been a PC industry; without a PC industry, there never would have been the hackers of the 1980s and 1990s. Without a military-industrial complex, which funded most of the computer research in the 1950s, 1960s, and 1970s, it is likely there would never have been a PC or a hacker culture. One cannot hope to understand the hacker in his or her modern incarnation without understanding at least a little bit about where the PC originated.

Steven Jobs and Steven Wozniak, inventors of the first mass-marketed PC, get a lot of mileage out of the story that they began as hackers, or, more specifically, as phone phreaks. Phone phreaks are to telephones what hackers are to computers — they possess a basic understanding of how the phone system works and as a result can do things such as place long-distance calls for free. One of the first phone phreaks was John Draper, aka Captain Crunch, who took his handle from his discovery that in the early 1970s the whistle that came with Capt'n Crunch cereal sounded the tone (2600 Hz) that would allow one to take control of the phone line and place long-distance calls for free. Not long after that, this discovery was harnessed in a more technical manner through what became known as a “blue box,” a small electronic device that would emit the 2600 Hz tone.

In contrast to Draper, Jobs and Wozniak learned about blue-boxing from a 1971 *Esquire* article²⁰ and built their own box. They not only used the boxes to make free calls but also went on to sell them to students in the Berkeley dorms.²¹ In short, Jobs and Wozniak copied a device someone else had originated and then *sold* those copies. This is a pattern Jobs and Wozniak would follow in building the first Apple. In contrast to the long-standing hacker ethic that freely distributed information and knowledge and resisted the impulse toward commodification, Jobs and Wozniak openly embraced it.

Before there was the Apple, however, there was the Altair. The Altair was the very first PC. It came in parts, cost between four hundred and five hundred dollars, and had to be built (and not just assembled, but soldered) by the end-user. These limitations meant that PCs were

limited to computer hobbyists, and most of those hobbyists belonged to the first generation of old-school hackers. These were computer enthusiasts who did everything from soldering the wires to programming the machine. One of those enthusiasts was Bill Gates. Gates, along with his friend Paul Allen, began writing the first language for the Altair, what would become Altair BASIC. BASIC, however, was not Gates and Allen's invention, but rather a programming language that had been put into the public domain a decade earlier by Dartmouth researchers Thomas Kurtz and John Kemeny. Gates and Allen's contribution was making BASIC, a language designed for a large mainframe computer, run on the Altair.

Almost all of the software for the Altair was written by hobbyists, who routinely shared their programs with other hackers at meetings of computer clubs across the country. The sharing of information became one of the central tenets of the hacker ethic, and this would be the central organizing principle of one of the first computer clubs, the Homebrew Computer Club of Menlo Park, California. Later, as the early pioneers of hacking would make their mark in the corporate world, a new ethic would emerge that would change everything. The transformation was intensified as the process of commodification increased. With commodification, the earliest computer hackers, those who had built their computers themselves and shared every tidbit of information that might help to improve their machines or programs, were in competition with each other, fighting to create and to maintain market share. The result was dramatic — competition in the marketplace “retarded Homebrew’s time-honored practice of sharing all techniques, refusing to recognize secrets, and keeping information going at an unencumbered flow. . . . All of a sudden, they had secrets to keep.”²² That transition marks the dividing point between the old-school hackers of the 1960s and 1970s and the new-school hackers of the 1980s and 1990s.

The Hacker Imagination: From Sci-Fi to Cyberpunk

It should come as no surprise that the hacker tradition is grounded in the literature of science fiction and fantasy. Internet culture has

its roots in the first e-mail discussion list that emerged early in the life of ARPAnet (the precursor of today's Internet). That list was SF-LOVERS, a list of people devoted to the discussion of science fiction. But like the shift from the 1960s hackers to the 1990s hackers, sci-fi literature also underwent a radical shift. The 1960s hacker's inspiration was found in the literature of Isaac Asimov, Philip K. Dick, Norman Spinrad, or Harlan Ellison, all writers who depicted a future of possibility, who wrote cautionary tales. These tales usually begin in an unfamiliar world, one that isn't your own but threatens to be. Often set in the future, but not too far in the future, these novels present anything from an alternative history to a fantasy world where the strange and unusual are commonplace. For example, Philip K. Dick's 1969 novel *Ubik* begins with the line, "At three-thirty a.m. on the night of June 5, 1992, the top telepath in the Sol System fell off the map in the offices of Runciter Associates in New York City."²³ Dick's world is both familiar and strange—it contains elements we know and some we do not know. New York and all the familiar conventions are intact—time, date, and a business with a realistic name. Yet the description also points to the unfamiliar: in this new world telepathy is not only possible but has become a commodity. We are, in essence, introduced to a world of possibility that is familiar enough to be recognizable yet strange enough for us to take notice of the ways in which our future might change.

In comparison, the mainstay of the 1990s hacker was the literature of cyberpunk, represented by William Gibson and Jon Brunner. Their novels are predominantly dystopic, describing a battle that has already been fought and lost. Consider Gibson's opening to *Neuromancer*: "The sky above the port was the color of television, tuned to a dead channel."²⁴ Immediately we know that Gibson's vision is going to challenge the basic model of the cautionary tale. From the outset, what it presents is threatening, and the dystopia of Gibson's fiction is taken as preordained. The literature of cyberpunk so dominated the imagination of the 1990s hackers that, in many ways, they came to see themselves as antiheroes, based on the prototype of Gibson's characters and others. These characters live in a world defined, even in its geography, by information. In *Neuro-*

mancer, Gibson describes a world that the hackers of today have adopted as their own:

Home.

Home was BAMA, the Sprawl, the Boston-Atlanta Metropolitan Axis.

Program a map to display frequency of data exchange, every thousand megabytes a single pixel on a very large screen. Manhattan and Atlanta burn solid white. Then they start to pulse, the rate of traffic threatening to overload your simulation. Your map is about to go nova. Cool it down. Up your scale. Each pixel a million megabytes. At a hundred million megabytes per second, you begin to make out certain blocks in midtown Manhattan, outlines of hundred-year-old industrial parks ringing the old core of Atlanta.²⁵

In such an account, we can begin to see the manner in which the exchange of information and data begins to code more common points of reference. Not only have old, familiar locales (Manhattan, Atlanta) been recoded as data, but our entire way of seeing them has been transformed. The way we look at a map no longer is about distance but rather is about the density of information. The map not only is not the territory—it has lost its relationship to the territory in terms of representation. The map is no longer a scale model of space. What is represented is the “frequency of data exchange.” This is a revealing term. “Frequency,” a term of temporality, not location, is the first focus, and “exchange” is the second. What is lost in this “mapping” is any sense of place. Place is erased, making it possible for Gibson to write of “home” as a place without place, home as a system of exchange.²⁶

The literature of cyberpunk teaches, or reflects, the value of information both as data and as a social fabric, a medium of exchange, and a relational concept. Cyberpunk represents a world where information has taken over, and the literature provides a sense of the fears, dangers, anxieties, and hopes about that new world. As opposed to the earlier hackers who sought to “liberate” information, hackers of the 1990s see themselves as trapped by information. As a result, the “hacker ethic,” which took as its most basic tenet that

“information wants to be free,” needs to be radically transformed. While the spirit of the ethic remains intact, the letter of it must necessarily change. Where the hackers of the 1960s had a great deal of control over the information that they created and utilized, the hackers of the 1990s and the beginning years of the twenty-first century find themselves in a world so overwhelmed by information that control itself becomes the contested issue. Most important to the later hackers is the concept that information is now their home, and secrecy of information is the equivalent of confinement or prison. The original ethic is, for the most part, still intact, but its meaning, value, and application have been radically altered by the ways in which the world has changed.

The cyberpunk vision of the future has radically reshaped the vision of the latter-day hacker. In one document currently being circulated on the Internet, the “Declaration of Digital Independence,” one hacker describes the Internet as the next battleground for the regulation of information, and hence freedom. “It [the Internet],” he writes, “should be allowed to make its own rules. It is bigger than any world you can and can’t imagine, and it will not be controlled. It is the embodiment of all that is free; free information, friendship, alliances, materials, ideas, suggestions, news, and more.”²⁷ The hackers of the 1960s, inspired by the utopian science fiction of their day, saw the battle in terms of free information and felt encouraged by that literature to experiment, learn, and develop. In contrast, the hackers of today, with the dystopic vision of cyberpunk, see this battle as already lost and as something that can only be rectified by revolution. In part, this is the result of the increasing commodification of information, which has created a media that they describe as “the propaganda vending machine of today,” which, “as a whole, trip over themselves, feeding lies to the ignorant.”²⁸

These hackers hold that the commodification of information has led to an increasing investment of power in the media. Accordingly, they argue, a transformation has taken place. The media, who “enjoy the power of managing information” in an era of commodification of information, are no longer interested in the freedom of information, but, rather, are invested in the careful control and dissemination of information. As Paul Virilio argues, the commodification of infor-

mation has had an ironic effect: “[T]he industrial media have gone the way of all mass production in recent years, from the necessary to the superfluous. . . . [T]heir power to denounce, to reveal, to flaunt has been growing endlessly to the detriment of the now precarious privilege of dissimulation — so much so that currently the real problem of the press and television no longer lies in what they are able to show as much as in what they can still manage to obliterate, to hide.”²⁹ The point is not so much that the media do not break news or reveal secrets as much as it is that they are selective about which secrets can and will be revealed. As media power is increasingly consolidated, media outlets have a possessive investment in particular stories and a similar investment in keeping other stories quiet. The more centralized the media become, the more power they have to self-regulate what constitutes “news.”

Today’s hackers, who follow this dystopic vision, contrast the media’s approach to the management of information with their own sense of boundless curiosity. As the “Digital Declaration of Independence” illustrates: “Everyone has the need to know, the curiosity of the caveman who invented fire, but some have been trained like monkeys, not ever knowing it’s there. They simply accept things, and do what is expected of them, and this is sad. They are those who never fight back, and never open their minds. And they are, unfortunately, usually the governing bodies; the teachers, bosses, police, federal agents, congressmen, senators, parents, and more. And this, my friend, must change.”³⁰ Such a call to action defines the problem in terms of curiosity and recognizes that such curiosity is problematic within contemporary culture. Curiosity becomes dangerous and even subversive not to any particular group or organization but in principle. Curiosity is precisely what threatens secrecy, and in doing so, it challenges the economic structure, the commodification, of information.

Blurring the Lines between Old and New: *WarGames*, Robert Morris, and the “Internet Worm”

In 1988, the distinction between the old and the new hacker was clearly staked out. This difference was revealed through two fig-

ures: David Lightman, the protagonist from the film *WarGames* and the prototype for the new-school hacker; and Robert Morris, a quintessentially old-school hacker.

The new-school hacker was introduced into the popular imagination through the 1983 release of the film *WarGames*, featuring Lightman (played by Matthew Broderick) as a curious kid exploring computers and computer networks who unwittingly starts the U.S. military on the road to World War III by playing what he thinks is a game — global thermonuclear war.

WarGames opens, somewhat ominously, with a scene in which U.S. soldiers in missile silos are ordered to fire their weapons at the Soviet Union, beginning what they believe will be the third, and undoubtedly last, world war. The soldiers are uncertain as to whether the order is part of a training exercise or not. As a result, a large percentage of the soldiers, uncertain about the effects of their actions, choose not to fire their missiles. The orders are part of a simulation, designed to test U.S. military battle-readiness. Their failure results in the implementation of a new program, wherein humans are removed from missile silos and replaced by electronic relay switches and strategic decisions about nuclear warfare are to be made by a state-of-the-art computer named WOPR (pronounced whopper) — War Operations Planned Response. The machine is devoted to constantly replaying World War III in an effort to maximize the effectiveness of U.S. missiles and minimize U.S. casualties. In effect, its job is to figure out how to win a nuclear war.

The young hacker, David Lightman, stumbles across WOPR quite accidentally while searching for a game company called Protovision. Using a modem and computer program (since termed a *WarGames* dialer), the hacker scans every open phone line in Sunnyvale, California (in the heart of the Silicon Valley), looking for modem access to Protovision. He comes across a system and, through hours of research (principally learning about the system's designer, Stephen Falken) and hacking, gains access.

The initial game continues to run even after Lightman is disconnected from the system, and, as Lightman soon discovers, the computer is unable to distinguish the game from reality. In less than three days time, WOPR will calculate a winning strategy and fire its

weapons. In the meantime, Lightman is arrested. On the verge of being charged with espionage, he escapes custody and searches out the system designer with the help of his girlfriend, Jennifer Mack (played by Ally Sheedy), to convince him to help persuade NORAD (North American Air Defense command) not to believe the information that the computer is sending them. Falken, the system designer, has become a recluse since the death of his son, Joshua (whose name is the secret password that gains Lightman access and the name that Lightman and Falken use to refer to the WOPR computer). Eventually, after several incidents that put the world on the brink of a nuclear holocaust, Lightman manages to teach the computer (now Joshua) the meaning of futility by having it play ticktacktoe in an infinite loop sequence. The computer concludes, after exhausting the game of ticktacktoe and learning from it, that nuclear war is also unwinnable and that "the only winning move is not to play."

The film demonstrates a tremendous anxiety about technology, represented both by the missiles that threaten to destroy the United States and the Soviet Union and by the machines that control those missiles.³¹ The hacker, however, is represented in a more ambivalent manner. On the one hand, the harmlessness of Lightman's actions early on (using his computer to make faux airline reservations or even finding a bank's dial-in line) is made clear by his good nature and curiosity. That sense of "playing a game" is radically transformed when the machine he hooks up to (the WOPR) is unable to distinguish between a game and reality. On the other hand, Lightman and his understanding of "playing a game" (in this case ticktacktoe) ultimately are able to save the day.

In the film, the hacker is positioned as dangerous because he is exploring things about which he has little or no understanding. It is easy in a world of such great technical sophistication, the film argues, to set unintended and potentially disastrous effects into motion even accidentally. But equally important is the characterization of the hacker as hero. WOPR, above all else, is a thinking machine, an artificial intelligence, and that thinking machine needs guidance and instruction in its development. Technology is infantilized in the film (underscored by the use of the name of Falken's deceased son, Joshua), and the message of the film is that Lightman, the hacker, is

the most appropriate educator for the technology of the future. Not the generals, the system administrator, or even the system designer himself is able to teach the machine, but Lightman, the hacker, can. The hacker stands at the nexus between the danger and the promise of the future of technology.

Thus, the intersection of hacker culture and popular culture is clearly and conspicuously marked. With the release of *WarGames*, hacker culture had a national audience. That culture, however, was as much a product of the film (and the response to the film) as it was a reality. While there certainly had been a long history of hacking and phreaking that predated *WarGames*, the hacking community itself was small, exclusive, and rather inconspicuous. With *WarGames* that all changed. As Bruce Sterling describes it, “with the 1983 release of the hacker-thriller movie *WarGames*, the scene exploded. It seemed that every kid in America had demanded and gotten a modem for Christmas. Most of these dabbler wannabes put their modems in the attic after a few weeks, and most of the remainder minded their Ps and Qs and stayed well out of hot water. But some stubborn and talented diehards had this hacker kid in *WarGames* figured for a happening dude. They simply could not rest until they had contacted the underground — or, failing that, created their own.”³² To varying degrees, hackers themselves admit to this, although none would probably state it in precisely Sterling’s terms. One of the primary differences between the film’s depictions and Sterling’s is that the film *WarGames* has absolutely no representation, or even suggestion of, an underground.

Undoubtedly, the film had a greater impact on hacker culture than any other single media representation. Hackers such as Shooting Shark and Erik Bloodaxe, in discussing their early influences, both confess (somewhat reluctantly) that the film had a major impact on them. “Embarrassing as it is for Erik,” his Pro-File reads, “*WarGames* really did play a part in imbedding the idea of computer hacking in his little head. (As it did for hundreds of others who are too insecure to admit it.)”³³ Shooting Shark’s chagrin is equally obvious, “Worse yet, ‘*WarGames*’ came out around this time. I’ll admit it, my interest in hacking was largely influenced by that film.”³⁴ Many hackers took their handles from the film, including “David

Lightman” and a whole spate of Professor Falkens or, more often, Phalkens, named for the computer genius who invented the WOPR. The film has even been held as the inspiration for one of the most serious instances of hacking in the 1980s, the exploits of a West German hacker who infiltrated U.S. military computers and sold U.S. government information to the Soviets.³⁵ The fact of the matter is that while many teens may have been intrigued by the possibility of breaking into high-security military installations, that aspect was never glamorized in the film. More likely, what intrigued young hackers-to-be was an earlier scene, where Matthew Broderick logs on to the local school’s computer using a modem from his PC at home and changes his grades in order to avoid having to take classes in summer school. The lure for the young hacker was never with starting World War III, but rested, instead, with the ability to make local conditions (for example, school work) more tolerable.

This narrative presents the good-natured, well-intentioned hacker innocently wreaking havoc as a result of his explorations. If Lightman is the introduction of the hacker to the popular imagination, Robert Morris would be the turning point in the perception of what kinds of threats hackers pose to society.

Perceptions of who and what hackers are underwent another transformation in the late 1980s, and that moment can be marked quite clearly in the popular imagination by a single case — the day the Internet shut down, November 8, 1988. Although the Internet was only a fraction of the size that it is today, it was an information infrastructure that had grown large enough that many government agencies (not the least of which was the military) and especially colleges and universities had come to rely on it. *WarGames* had provided the fantasy, the cautionary tale, but it all had ended well. There had been no disaster, and, even though the hacker had created the problem himself, it was also the hacker who saved the day, teaching the computer the lesson of futility by giving it a game (ticktacktoe) that it couldn’t win, by becoming the educator of the technology of the future. The hacker, in the old-school tradition, saved the day by rendering technology itself (in that case WOPR) benign. Such was not the case with Robert Morris.

In 1988, Morris launched what has come to be known as

the “Internet worm,” a computer program that transmitted itself throughout the Internet, eating up an increasing number of computing cycles as it continually reproduced itself. Morris’s intent was to have the worm reproduce at a rate that would allow it to continue throughout the system, unnoticed for years. With each system it entered, it would exploit a security hole, discover user passwords, and mail those passwords back to Morris. Morris, however, made a small error that had a great effect, and the worm reproduced itself at a rate much faster than he had anticipated. Rather than telling the worm to stop running when it encountered a copy of itself, which would have stopped it from spreading, Morris coded the worm so that one in every seven worms would continue to run even if other copies of the worm were found on the machine. The ratio was too high “by a factor of a thousand or more,” and Morris’s worm replicated itself at an astonishing rate, essentially eating up computing cycles at such a rate that the machines could do nothing but run Morris’s program.³⁶

Morris’s worm spread throughout the nation’s computer systems at an alarming rate. The day after the worm was released, a significant portion (no one is really sure of the exact number) of those computer systems hooked to the Internet had ground to a dead stop. The only remedy was to disconnect from the network and wait for the experts working around the country to find a way to counteract the program.

What makes this case special — apart from the fact that it focused national attention on issues of computer security, Net vulnerability, and the degree to which we have come to depend on computers in our daily lives — was the ambivalent stature of the programmer himself. By all rights, Robert Morris was an old-school hacker. A computer-science graduate student at Cornell, he had spent most of his college career (and early years) working with computers, devising clever workarounds for difficult problems, and engaging with his special fascination, computer security. The son of Bob Morris, the chief scientist at the National Security Agency’s National Computer Security Center, Robert learned the ins and outs of computer and system security at a very early age and possessed a natural talent for finding system bugs and holes he could exploit to gain access.³⁷

Even the worm itself was never intended as anything more than an exposition of the flaws in UNIX system design.

Morris's prank, however, was extraordinarily similar to the most malicious activity that was being attributed to the new breed of hackers, the computer virus. The difference between a worm and a virus was hotly debated within the computer community.³⁸ Worms, which are independent programs that move under their own power, are generally considered to be beneficial things. First developed by researchers at Xerox, worms were programs that would run throughout a system performing useful tasks. The name is taken from John Brunner's 1975 novel, *Shockwave Rider*, where the protagonist feeds a "tapeworm" into the government's computer, as a means to counter government surveillance. The act, in Brunner's case, is one of heroism in the face of an oppressive, tyrannical government. In that sense, worms carry with them the connotation of performing a useful function, usually in line with the traditional hacker ethic. Many were reluctant, it seems, to label Morris's program a worm because of the positive connotations that the term carried.

Viruses, in contrast, are generally considered to be malicious. The connotations of viruses as sickness, illness, and even death (particularly in the age of AIDS and Ebola) provide an interesting counterpoint to the discussion of the "Internet worm." Many who argued for the classification of Morris's program as a virus did so on the ground that it was harmful and caused great damage. The community was split both on how to refer to the program and on its ethical implications. The line between the old school and the new school had begun to dissolve.

Morris's worm program spurred a great deal of speculation about his intentions and influences (Morris refused to talk with the press) and a great deal of debate over the threat that technology posed to society if it fell into the wrong hands. As Katie Hafner and John Markoff put it, "it also engaged people who knew nothing about computers but who were worried about how this new technology could be used for criminal ends."³⁹ In the popular imagination, the line between the old and new hacker was already fuzzy, and Morris's worm program only made the distinction fuzzier.

A second drama was also being played out around anxieties

about technology. Those anxieties coalesced around the notions of secrecy and technology and were reflected in popular culture's representations of hackers.

For example, films about hackers almost always deal with the question of secrecy. If we are to take *WarGames* as the prototypical hacker film, it is easy to see exactly what this means. *WarGames* begins with David Lightman trying to break into Protovision in an effort to play the latest games before they are publicly released. In essence, Lightman seeks to break Protovision's corporate secret in order to have access to "secret" games. As the name suggests, Lightman wants to see something before anyone (or, more to the point, everyone) else does. While searching for Protovision's phone number, Lightman comes across a more interesting system, one that "doesn't identify itself." Thinking that this is Protovision's dial-in number, Lightman attempts to hack into the system, even after he is warned by two system programmers that the system is "definitely military."

The crucial point about the basic theme underwriting the film is that these two cultures of secrecy, Protovision's and NORAD's, are virtually identical. The structure of *WarGames* depends on our understanding and acceptance of the confusion of a corporate computer-game manufacturer's notion of secrecy with that of NORAD. In other words, corporate and military secrets are, at some level, indistinguishable. In the films that followed *WarGames*, this theme was renewed and expanded upon. In *Sneakers* (1992), the confusion between corporate and government secrecy is complete when it is revealed that the only real governmental use for the "black box" agents have been sent to recover is to snoop on other U.S. governmental departments, rather than on foreign governments. What we witness throughout the film, however, is that as a corporate tool, that black box is capable of everything from corporate espionage to domestic terrorism. Indeed, the project "SETEC ASTRONOMY" which is at the center of *Sneakers*, is an anagram for "TOO MANY SECRETS." And while it is a government project (funded by the NSA in the film), it is revealed in the film's epilogue that the box will not work on other countries' cryptographic codes, only on those of the United States. Accordingly, the message is clear: domestically, there are to be "no more secrets" kept from those in power.

In two films of the 1990s, *Hackers* (1995) and *The Net* (1995), secrecy plays a major role as well. In these films, however, secrecy is what allows criminality to function in both the government and corporate worlds. In the case of *Hackers*, two employees (one a former hacker) of a major corporation are running a “secret worm program” to steal millions of dollars from the corporation. They are discovered when a hacker unwittingly copies one of their “garbage files” that contains the code for the worm’s program. The same plot is played out in *The Net*, but from a governmental point of view. Angela Bassett (played by Sandra Bullock) accidentally accesses and copies secret governmental files that reveal wrongdoing on the part of governmental officials, again demonstrating the manner in which the culture of secrecy is able to hide and allow for a deeper sense of criminality to ferment and function. In both cases, hackers, by violating the institutions’ secrecy, expose criminality by enacting criminality. The message from the later films is that secrecy creates a space for the worst kinds of criminality, which, because of the culture of secrecy, can only be exposed by another type of criminality — hacking.

Hackers of Today

As Steve Mizrach has noted, the split between the hackers of the 1960s and those of today is cultural and generational rather than technological:

The main reason for the difference between the 60s and 90s hackers is that the GenXers are a “post-punk” generation, hence the term, “cyberpunk.” Their music has a little more edge and anger and a little less idealism. They’ve seen the death of rock n’ roll, and watched Michael Bolton and Whitney Houston try and revive its corpse. Their world is a little more multicultural and complicated, and less black-and-white. And it is one in which, while computers can be used to create beauty, they are also being used to destroy freedom and autonomy. . . . [H]ence control over computers is an act of self-defense, not just power-hunger. Hacking, for some of the new

“hackers,” is more than just a game, or means to get goodies without paying for them. As with the older generation, it has become a way of life, a means of defining themselves as a subculture.⁴⁰

Born in the world that the 1960s hackers shaped, this new generation has been jaded precisely by the failure of the old-school hackers to make good on their promises. Technology has not been rendered benign, and information, while still anthropomorphized and personified, reveals that we are more like it — confined, coded, and organized — than information is like us. The cautionary tales, like Asimov’s, that guided the 1960s hackers have been replaced with tales of a dystopian cyberpunk future that features technology no longer in the service of humankind, but humankind fused with technology through cybernetics, implants, and technological modifications to the body.

There seem to be two reasons for the shift. The first is that the rate of technological growth has outstripped society’s capacity to process it. A certain technophobia has emerged that positions technology as always ahead of us and that produces a fear that is embodied by the youth of contemporary culture doing things with computers that an older generation is unable to understand. Hacking promotes fear, but it is about a contained kind of fear, one that is positioned as a form of “juvenile delinquency” that these youth will, hopefully, grow out of. In that sense, hackers emerge as a type of “vandal,” a criminal who is often malicious, who seeks to destroy things, yet is terribly elusive. The threat, like the technology that embodies the threat, is decentralized, ambiguous, and not terribly well understood, but it doesn’t need to be. We feel we can trust our information networks, for the most part, the same way we can trust our trains and buses. Occasionally, someone may spray paint them, flatten a tire, or set a fire on the tracks, but these things are inconveniences, not disasters. Hackers pose a similar type of threat: they may deface the surface of things, but the underlying faith in the system remains intact. Just as one does not need to understand how internal combustion engines work to trust that a car will function properly, one does not need to understand how information networks function in order to

use them. The second reason for a fear of hackers is the result of a displacement of anxiety that the hackers of the 1960s have identified — namely, the increasing centralization of and lack of access to communication and information. This new generation of hackers has come to represent the greatest fear about the 1960s dream of free and open information. John Perry Barlow, cofounder of the EFF (Electronic Frontier Foundation), wrote in the organization's manifesto, titled "Crime and Puzzlement," about his first interactions with two members of the New York hacking collective Masters of Deception. These two hackers, Phiber Optik and Acid Phreak, in an effort to demonstrate their online skills, threatened Barlow in one of their first encounters in a most unusual way:

Optik had hacked the core of TRW, an institution which has made my business (and yours) their business, extracting from it an abbreviated (and incorrect) version of my personal financial life. With this came the implication that he and Acid could and would revise it to my disadvantage if I didn't back off.⁴¹

What is unusual about this threat is the manner in which it employs the vision of the 1960s hacker with a completely inverted effect. The threat is the removal of secrecy — a true freedom of information. It is important to note that TRW's report was not something that Barlow had ever consented to, nor was it something that he had any control over. In making it free, accessible, and open, the hackers posed the greatest threat — the ability to change the unchangeable, to access the secret, and to, in the process, disrupt a significant portion of one's life. If nothing else, the importance of the secrecy of information is documented (not coincidentally) by the hacker in the most dramatic fashion. Indeed, downloading TRW reports is a common trick that hackers will employ to startle, scare, and even intimidate media personalities, interviewers, and, in some cases, judges and lawyers. What Barlow identifies, with comic overtones, illustrates precisely what the implications of this reversal are. "To a middle-class American," Barlow writes, "one's credit rating has become nearly identical to his freedom."⁴² That is, freedom relies on secrecy. The culture of information that the 1960s hacker feared has come to pass. In a kind of

Orwellian doublespeak, secrecy has become freedom, and the need for security (through implicit distrust of others) has become trust.

Old-school hackers, such as Clifford Stoll, a Berkeley astronomer and system administrator, put forth the thesis that computer networks are and should be built on trust, a principle that seems in line with the 1960s hacker ethic. When confronted with a hacker who explained that he had hacked into Stoll's system to "show that your security isn't very good," the latter replied, "But I don't want to secure my computer. I trust other astronomers."⁴³ However, hackers of the 1990s argue that it is precisely this sort of argument that illustrates the hypocrisy of the 1960s hackers, many of whom have become rich in the computer revolution precisely by betraying the principles of openness, access, and freedom that they argued for as their ethic. The most interesting example is, perhaps, one of the most illustrative as well. As Steve Mizrach argues, "[Steven] Levy rants about those greatest hackers who founded Apple Computer and launched the PC revolution — those same ex-phreaks, Jobs and Wozniak, who actually allowed their company to patent their system hardware and software!"⁴⁴ To this day, Apple has been extremely successful in keeping both its hardware and software proprietary.

In essence, today's hackers argue, with a great deal of justification, that the hackers of the 1960s have become their own worst nightmare, the keepers of the secrets, those who block access to information and technology, all in the name of corporate self-interest. The new-school hacker, then, seems little more than the logical carryover from the earlier generation, a generation that spoke (and continues to speak) with such earnest commitment to the "good-old days" that they are unable to see how their own ethic implicates them in precisely that which they so fervently disavow. The connection, which the 1960s hackers are so loathe to make, is, in many ways, undeniable: "Indeed, the 90s hackers pay a lot of homage to the first generation. They have borrowed much of their jargon and certainly many of their ideas. Their *modus operandi*, the PC, would not be available to them were it not for the way the 1960s hackers challenged the IBM/corporate computer model and made personal computing a reality." In that sense, today's hackers are the children of the 1960s hackers, and that connection is not lost on the younger

generation. In fact, today's hackers have inherited not only the tools from the older generation but much of their culture as well: "[T]heir style, their use of handles, their love for late-night junk food, are all testaments to the durability and transmission of 1960s hacker culture."⁴⁵

They have also inherited, for the most part, their ethic. That ethic has been transformed, undoubtedly, but so have the conditions under which that ethic operates. These conditions are, in many ways, the progeny of the 1960s as well. Exploring, which seemed so harmless to the 1960s hacker, was only harmless because the culture of secrecy had not fully taken hold. As Barlow describes his in-person meeting with Phiber Optik, he "encountered an intelligent, civilized, and surprisingly principled kid of 18 who sounded, and continues to sound, as though there's little harm in him to man or data. His cracking impulses seemed purely exploratory, and I've begun to wonder if we wouldn't regard spelunkers as desperate criminals if AT&T owned all the caves."⁴⁶ This was the same hacker who had just days before threatened Barlow with a revision of his TRW credit report that threatened to destroy him financially.

How are we to explain this seemingly split personality? There are several reasons the current hacker comes across so brazenly, not the least of which is the proliferation of media in which hackers have begun to appear. While the 1960s hacker was confined to talking, predominantly, to other hackers, today's hacker is online in a world where there are few aspects of daily life that are not controlled or regulated by computers. Where computers were a novelty in the 1960s, today they are a desktop necessity. As computers entered the popular imagination, the hacker came along and was transformed with them.

A primary difference between the hackers of the 1960s and those of today rests with the fact that the latter are, for want of a better term, "media ready." For example, *CuD* (*Computer Underground Digest*), an online publication that tracks news of the underground, published an essay in response to a Fox News story on the "Hollywood Hacker" that eventually led to his arrest and prosecution. Although the essay does take up questions of the specifics of how computer raids are conducted, how warrants are obtained, and so on, the first two issues of concern, listed in the section "Why Should

the CU Care?" are as follows: "1. The role of the media in inflaming public conceptions of hacking seems, in this case, to exceed even the cynical view of sensationalistic vested interest." Of equal concern is the "hacker hyperbole" that accompanied the Fox news report. "2. A second issue of relevance for the CU is the definition of 'hacker.'" What is at stake for the computer underground is the very control of the term "hacker" and what constitutes "hacking."⁴⁷

To say that there is an awareness among hackers of how they are portrayed in the media would be a drastic understatement. Speaking of Phiber Optik and Acid Phreak, Barlow describes the phenomenon: "They looked about as dangerous as ducks. But, as *Harper's* and the rest of the media have discovered to their delight, the boys had developed distinctly showier personae for their rambles through the howling wilderness of Cyberspace." These personae are not merely inventions of the media, but are formed in a kind of cooperative venture between the media and the hackers themselves. "Glittering with spikes of binary chrome," Barlow writes, "they strode past the klieg lights and into the digital distance. There they would be outlaws. It was only a matter of time before they started to believe themselves as bad as they sounded. And no time at all before everyone else did."⁴⁸

Books, articles, newspaper reports, films, and TV documentaries have tracked, with varying degrees of accuracy, the exploits of some of today's more high-profile hackers. One hacker, Kevin Mitnick, who has led authorities on several nationwide manhunts as a result of his hacking exploits, has been the subject of three books that detail his crimes and exploits. Others, such as Robert Morris, Phiber Optik, Kevin Poulsen, the members of the Legion of Doom, and the Masters of Deception have all been featured in books that range from the journalistic to high drama.

These new hackers have captured the spotlight in a way that the hackers of the 1960s never did. Old hackers captured attention in complete anonymity, for example, by sneaking into the Rose Bowl prior to the game and substituting the cards held by fans during the game so that they read "Cal Tech" rather than "Washington" (as they were supposed to). It made no difference to the hackers that Cal Tech wasn't even playing that weekend.⁴⁹ In contrast, in 1988, when AT&T suffered a major failure in long-distance telephone ser-

vice that was found to be the result of a software glitch, reports immediately circulated that the interruption of service was the result of a hacker in the New York area who had broken into AT&T's computer system as a protest against the arrest of Robert Morris.⁵⁰ The media, as well as the public, have learned to expect the worst from hackers, and as a result, hackers usually offer that image in return, even if their own exploits are no more than harmless pranks or explorations grounded in curiosity.

If hacking is about imagination, then the reasons hackers hack are probably as numerous as the hackers themselves, and the means by which they accomplish their tasks range from the highest end of the technological spectrum to the lowest. But if we are to understand what hacking is and who hackers are, we need to separate out the people from the machines. To divorce hacking from its technical aspects, however, is not to divorce it from technology altogether. Hacking is about technology; arguably it is about nothing but technology. Hackers and hacking constitute a culture in which the main concern is technology itself and society's relationship to the concept of technology. Accordingly, I adopt the term "culture of technology" as a way to understand the cultural implications of technology from the hackers' point of view as well as from a broader cultural standpoint.

Hackers themselves rarely, if ever, talk about the tools they use. Indeed, their activity demonstrates that computer tools are above all mere vehicles for activity. In many cases, today's hackers utilize common UNIX servers as their goals, targets, and mechanisms for hacking — that is, they utilize other people's resources as a means to accomplish their goals. Even the most high-powered PC is little more than a dumb terminal that allows the hacker to connect with a more powerful corporate or university machine, an act that can be accomplished just as effectively with a fifteen-year-old PC or VT100 terminal as it can with the fastest, highest-end multimedia machines on the market today.

Hacking is not, and has never been, about machines, tools, programs, or computers, although all of those things may appear as tools of the trade. Hacking is about culture in two senses. First, there is a set of codes, norms, values, and attitudes that constitute

a culture in which hackers feel at home, and, second, the target of hackers' activity is not machines, people, or resources but the relationships among those things. In short, hacking culture is, literally, about *hacking* culture. As culture has become dependent on certain types of technology (computers and information-management technology, particularly), information has become increasingly commodified. And commodification, as was the case with the first personal computer, is the first step in the revaluing of information in terms of secrecy. As Paul Virilio maintains, along with commodification comes a new way of valuing information — in “the maelstrom of information in which everything changes, is exchanged, opens up, collapses, fades away, gets buried, gets resurrected, flourishes, and finally evaporates in the course of a day,” duration no longer serves as an adequate means of valuation. Instead, he argues, “speed guarantees the secret and thus the value of all information.” Accordingly, American, and perhaps all of Western, culture's relationship to information has been undergoing radical change, moving from a culture that values duration to one that values secrecy. That transformation also marked the moment of emergence of the hacker, a moment Virilio situates as a “data coup d'état” that originated with the “first military decoders to become operational during the Second World War.” It was those machines, the “ancestors of our computers and software systems,” that produced the merging of “information and data processing” with the “secret of speed.”⁵¹ And, indeed, Virilio is right in the sense that World War II produced the first machines that were capable of what we currently think of as cryptography and that were, perhaps, the beginning of the union of information and speed to produce secrecy. The employment of machines that were, in essence, rudimentary computers made complex coding and decoding efficient. This allowed for the production of codes that were much more complex and, therefore, much more difficult to break. As speed and information merge, secrecy becomes an increasingly important component of the culture in which we live. But such secrecy is precisely what hacker culture abhors.

Secrecy is not limited to encryption schemes but begins, as Virilio points out, with the process of commodification. In 1975, when hobbyists were busily programming the Altair, coding was done on

paper tape and had to fit in 4K of memory. As noted earlier, one of the first successes was Altair BASIC, a programming language written by two college students, Paul Allen and Bill Gates, that would allow others to develop software for the Altair. The difference between Allen and Gates's Altair BASIC and just about every other program written for the Altair was that Allen and Gates sold their program, rather than giving it away. This difference was enormous, since for computer hobbyists, the question was never one of profit, but one of access. Dan Sokol, the person who had obtained and copied the original version of Allen and Gates's BASIC, distributed the program at the next meeting of the Homebrew Computer Club and "charged what in hacker terms was the proper price for software: nothing. The only stipulation was that if you took a tape, you should make copies and come to the next meeting with two tapes. And give them away."⁵² In no time, everyone had a copy of Allen and Gates's program. Bill Gates responded by sending an "Open Letter to Hobbyists," which was published both in the Altair users' newsletter and in the Homebrew Computer Club newsletter. "As a majority of hobbyists must be aware, most of you steal your software," Gates wrote, accusing hobbyists of being "thieves." "What hobbyist," Gates continued, "can put 3 man-years into programming, finding all the bugs, documenting his product and distributing for free?"⁵³

In essence, Allen and Gates treated the BASIC interpreter as a secret that could be purchased. Most other hackers didn't see it that way. And for them, ownership was precisely what was at stake. To violate the principle that "computer programs belonged to everybody" undercut every tradition of programming.⁵⁴ For the first generation of hackers, programming meant passing your work on for others to rewrite, rethink, debug and, generally, improve on. Secrecy and ownership, even at the level of commodification, made that impossible.

This sense of secrecy that developed along with the evolution of the PC changed the climate in which hackers operated. Part of the transformation from duration to speed is also a distancing of information. As information is made secret, language adapts, and, increasingly, language reflects the need for secrecy. Accordingly, as

the technology of language accommodates the possibility of secrecy, it too grows more distant. The result is not simply a feeling of being misunderstood or of alienation. Rather, this distance produces a more radical sense of being out of sync with the world, insofar as while one may speak or be able to speak the language of the world, it is not your first language; it is not your home language. In a culture of technology, the technology of language and the language of technology itself become more distant. But, for hackers, a subculture has emerged where the language of technology has taken over, where the language of technology is not distant but immediate. It is in this space that the hacker feels "at home." For example, Warren Schwader, after spending over eight hundred hours working on a program, "felt that he was inside the computer. . . . His native tongue was no longer English, but the hexadecimal hieroglyphics of LDX #0, LDA STRING,X JSR \$FDF0, BYT \$0, BNE LOOP."⁵⁵ Hacker culture emphasizes the degree to which technology defines culture. For hackers, the process of hacking exposes the manner and way in which culture relies on technology and the ways in which technology is constitutive of culture itself. In this sense, technology is the hacker's home culture, and, as a result, the hacker is at home speaking the language of technology.

While the hacker may feel at home in the language of technology, the evolution of secrecy has nonetheless distanced the technology of language. It is in this sense that hackers, even in a culture in which they feel at home, take hacking culture, which is to say the culture of secrecy, as their goal. Secrecy, in any form, is profoundly alienating. In fact, one of the basic conditions of secrecy is that one can never feel at home in relation to the secret. In order to remain a secret, information must be distanced through the technology of language, which can range from silence to commodification to patents and copyright to encryption. As information becomes more secretive, language itself become more inaccessible. Even the use of something as basic as an acronym illustrates the manner in which language can be made distant. In a visit to the doctor's office, the transformation from "central nervous system" to CNS not only marks a change in the language that doctors and nurses use but serves to distance patients from the discourse about their own

bodies. Companies, organizations, industry, and professions all develop coded languages that produce a sense of “home” for those who understand them and a sense of alienation or difference for those who do not.

For the hacker, the question is broader in scope. In contemporary culture, technology is colonizing language so rapidly that it is becoming the *lingua franca* for society. The hacker takes culture as his or her object, revealing how language operates as a technology and what the implications and effects are of the incorporation of secrecy into the relationship between language and technology.

Technology is the genesis of secrecy, so it is not surprising to find that the technology of language is already prepared to deal with hackers’ assault on the emergence and growth of secrecy in relation to the discourse surrounding computers and computer networks. Perhaps the most crucial metaphor is that of the host and its relation to a sense of home. The dynamics of this construction renders the relationship between hackers and the broader social culture transparent and gives us a trope upon which we can start to play out the meaning of “hacking culture.” The notion of the host implies the existence of a guest, a stranger who is met with either a sense of hospitality or a sense of hostility. Accordingly, the host must determine the threat of the other — if he or she comes as a guest, the other is met with hospitality; if she or he comes as an enemy, the other is met with hostility.⁵⁶ How one determines how the other is met is often a proprietary piece of information — a letter of introduction, a secret handshake, or a password. The metaphor of the host can also imply a discourse of infection within the language of computer networks themselves. The host serves as the basis for infection and gives rise to the notion of computer viruses: “The word virus,” one computer security book explains, “is a biological term pertaining to infections submicroscopic nucleo-proteins known mostly for their ability to invade a host cell, alter its DNA to produce more of its own nucleo-proteins, and finally, release these new versions of itself to invade surrounding cells. If you are to make an analogy of a computer virus to that of one in the world of biology, most of the properties of the stages or phases of a biological virus are identical to those of a computer system.”⁵⁷ Although discussions of viruses

are a fairly recent phenomenon, it seems the discourse of computer networks was already prepared for them. The language of computer systems has always relied on the tropes of hosts and guests, of users, of visitors and invaders. (For example, a host is any network computer that is “able to receive and transmit network messages,” and “guest” accounts are given to users whose identities are unknown or unconfirmed, that is, to strangers.)⁵⁸ The difference between the guest and the enemy is established by the knowledge of a secret, usually in the form of a password.

These discourses suggest that even the language where hackers might feel most at home, the language of computers, computing, and networks, was already set up to presume them as outsiders. The language where they should be most at home had already defined them as strangers, as outsiders, as invaders — they were, by definition, those who did not possess the secret.

The Ubiquity of the PC; or, Who Do You Want to Hack Today?

The strangeness that computer hackers feel is somewhat tempered today by the boom of the World Wide Web and the ubiquity of the PC. As technological savvy began to be associated with wealth in the wake of the Silicon Valley gold rush of the 1990s, representations of hackers became further bifurcated, and the essential curiosity and ethos of hacker culture were overwhelmed by the commercialization of hacking and hacker culture (the spirit of using technology to outsmart the system). This was brought about, on the one hand, by heightened images of criminality and, on the other hand, by several high-profile hacks and news stories that pushed hacking to the front pages of newspapers across the country. The fact of the matter is that as PC culture has become increasingly widespread, hacker culture has itself become increasingly divided. One set of divisions is between those who call themselves “white-hat” hackers, hackers dedicated to improving system security by seeking out flaws and finding ways to repair them, and “black-hat” hackers, who seek out flaws in order to exploit them. As hackers have grown up, left high school or college, and had to face things like mortgages, families, and

jobs, many of them have turned to work as “security professionals” or as system administrators with responsibilities that include keeping hackers out of their own systems.

A second division, however, has become equally important within the community, the split between hackers and what have become known as “script kiddies.” A “script kiddie” is someone who hacks, usually using someone else’s prewritten hacking script or program, without really understanding what he or she is doing. As a result, one can think of at least two distinct levels at which hacking occurs.

For example, on February 25, 1998, the Pentagon announced that it had been hacked (an admission that sparked speculation that the announcement was politically motivated, speculation that was confirmed when Janet Reno announced plans for a \$64 million center to help secure the nation’s government and military networks against “cyberterrorism” and online attacks two days later). The incident allowed hackers to clarify among themselves and in the media the difference between “true hacks” and “derivative hacks.” A “true hack,” the most sophisticated form of hacking, means finding and exploiting a security hole that was previously unknown. It is, for hackers, a discovery of the first order. “True hacks” are the result of understanding how things work (or, oftentimes, don’t work) and taking advantage of those flaws, oversights, or errors in an original way. This level of hacking requires intimate knowledge of computers, programs, and computer languages. These hacks are often discovered, reported, and patched by hackers themselves without ever using them to compromise some one else’s computer or security. The achievement is in the process of discovery, exploration, and knowledge. Hackers who make, and are capable of making, such discoveries represent a very small percentage of the culture. This segment of the community came of age in the late 1990s, finding it much easier to make their reputations by publicly documenting security holes than by exploiting them.

At the lowest level of sophistication is the “derivative hack,” which is simply the codified form of a true hack. Once a security hole is discovered, hackers write programs or scripts that allow the hack to be automated and run by just about anyone. No specialized programming knowledge is needed, and often the program will

come with instructions for use. These programs are widely available on the Net and can be downloaded and used by anyone with the inclination and even a basic understanding of how computer operating systems, such as UNIX, work. These hacks require the hacker to be able to match his or her tools to the job — knowing which machines have what bugs or holes and exploiting them. A smart kid with a rudimentary knowledge of computers can create a formidable arsenal in a matter of weeks using these programs and scripts. While these programs can be used to learn about systems and discover how things work, more often they are utilized by people who don't take the time to learn what they do. Hackers who spend the time exploring and learning about systems generally find these kinds of hacks unimpressive. One hacker, Oxblood Ruffin of the cDc, described them as “applications hackers” who do nothing more than “download cracking and hacking utilities and start running them on their machines and all of a sudden they find out that they can break into systems.” These hackers often run programs with little or no idea of how the programs work or what their effects will be. As Oxblood Ruffin explains it, “the problem really is that they don't understand what they are doing.” As a result, these programs are occasionally run with unintended consequences and cause accidental damage to systems, giving hackers a bad name.

From a hacker's point of view the attack on the Pentagon falls squarely in the second camp, a “derivative” or “applications” hack. At least one of the bugs used to attack government computers had been available since November 1997 and was available on the Net. The question from a hacker's point of view becomes, Why didn't the government bother to patch these widely known security holes in its systems? Hackers believe that your security should be comparable to the value of the information you want to protect, and leaving gaping security holes is tantamount to an invitation to enter the system.

What is rarely discussed is that true hacks, which do present a very real security threat, are almost always discovered for the purpose of *increasing* security. Hackers make their reputations by releasing these bugs and holes in basic “security advisories,” by publishing them in hacker journals, by posting them online at places like The L0pht (a Boston hacker collective), or by publishing them on mailing

lists such as Bugtraq or RISKS Digest (the two most widely read security e-mail discussion groups).

Derivative hacks, on the other hand, are seen as nothing more than joyriding. In most cases, hackers see nothing inventive or particularly clever about breaking into a system using tools that someone else created. It is a bit like stealing an unlocked car with the keys in the ignition.

Even though the hackers who hacked the Pentagon's computer (which turned out to hold nonsensitive accounting records) presented no risk to national security, the story was reported and sensationalized in most of the mainstream media. Even online media were not immune (though some, such as *Wired News*, did a much better job of clarifying and reporting the issue).

For example, CNN.com didn't bother to verify the attacks or assess them in an independent way. Instead, they were reported as characterized by Pentagon officials, who admitted to knowing very little about computers themselves and who described the attacks as an "orchestrated penetration" and as the most "organized and systematic attack the Pentagon has seen to date." The report was certainly sensational, but hardly accurate.

In assessing the risk that such an attack posed, CNN.com reported Israeli prime minister Benjamin Netanyahu's description of the young Israeli teen accused of the attacks. When asked, Netanyahu called Analyzer "damn good... and very dangerous." In many ways Netanyahu's response was predictable and politically expedient, but it begs the question as to whether or not the prime minister of Israel is qualified to assess the quality of Analyzer's hacking abilities.

The figure of the hacker, at least since the movie *WarGames*, has been the source of a great deal of anxiety in contemporary culture. The hacker is the personification of technology, representing mystery and danger. The hacker is mysterious because he or she appears to work magic with computers, phone lines, modems, and codes, and she or he is dangerous for precisely the same reasons. The hacker is the figure upon which we can heap all of our anxiety about technology, and when the news media report a break-in at the Pentagon, all of our worst fears appear to be realized. By treating such events as

sensational, the media answer the question, How did an eighteen-year-old kid break into the Pentagon? by preserving the illusions of fantasy and science fiction and by playing on misperceptions and basic fear of the unknown: he did it by magic.

As a result, breaking into computers and the threat that hacked Web pages create to corporate images have come to symbolize the new age of hacking, though these feats are easily achievable by anyone who has a weekend to spare and the right Web page addresses to find the scripts they need to exploit security flaws. As Ira Winkler (a former NSA consultant and security expert) is fond of saying, "I could teach a monkey how to break into a computer." Hackers, however, are much more interested in finding security flaws and understanding how they threaten network and computer security. They are busy trying to understand how the system works, whether that be as a means to exploit it or to better understand it. Even as the phenomenon of hacking becomes more widespread, the core of the culture remains true to a basic set of beliefs.