# 0x100

## INTRODUCTION

The idea of hacking may conjure stylized images of electronic vandalism, espionage, dyed hair, and body piercings. Most people associate hacking with breaking the law and assume that everyone who engages in hacking activities is a criminal. Granted, there are people out there who use hacking techniques to break the law, but hacking isn't really about that. In fact, hacking is more about following the law than breaking it. The essence of hacking is finding unintended or overlooked uses for the laws and properties of a given situation and then applying them in new and inventive ways to solve a problem—whatever it may be.

The following math problem illustrates the essence of hacking:

> Use each of the numbers 1, 3, 4, and 6 exactly once with any of the four basic math operations (addition, subtraction, multiplication, and division) to total 24. Each number must be used once and only once, and you may define the order of operations; for example, 3 * (4 + 6) + 1 = 31 is valid, however incorrect, since it doesn't total 24.

The rules for this problem are well defined and simple, yet the answer eludes many. Like the solution to this problem (shown on the last page of this book), hacked solutions follow the rules of the system, but they use those rules in counterintuitive ways. This gives hackers their edge, allowing them to solve problems in ways unimaginable for those confined to conventional thinking and methodologies.

Since the infancy of computers, hackers have been creatively solving problems. In the late 1950s, the MIT model railroad club was given a donation of parts, mostly old telephone equipment. The club's members used this equipment to rig up a complex system that allowed multiple operators to control different parts of the track by dialing in to the appropriate sections. They called this new and inventive use of telephone equipment *hacking*; many people consider this group to be the original hackers. The group moved on to programming on punch cards and ticker tape for early computers like the IBM 704 and the TX-0. While others were content with writing programs that just solved problems, the early hackers were obsessed with writing programs that solved problems *well*. A new program that could achieve the same result as an existing one but used fewer punch cards was considered better, even though it did the same thing. The key difference was how the program achieved its results—*elegance*.

Being able to reduce the number of punch cards needed for a program showed an artistic mastery over the computer. A nicely crafted table can hold a vase just as well as a milk crate can, but one sure looks a lot better than the other. Early hackers proved that technical problems can have artistic solutions, and they thereby transformed programming from a mere engineering task into an art form.

Like many other forms of art, hacking was often misunderstood. The few who got it formed an informal subculture that remained intensely focused on learning and mastering their art. They believed that information should be free and anything that stood in the way of that freedom should be circumvented. Such obstructions included authority figures, the bureaucracy of college classes, and discrimination. In a sea of graduation-driven students, this unofficial group of hackers defied conventional goals and instead pursued knowledge itself. This drive to continually learn and explore transcended even the conventional boundaries drawn by discrimination, evident in the MIT model railroad club's acceptance of 12-year-old Peter Deutsch when he demonstrated his knowledge of the TX-0 and his desire to learn. Age, race, gender, appearance, academic degrees, and social status were not primary criteria for judging another's worth—not because of a desire for equality, but because of a desire to advance the emerging art of hacking.

The original hackers found splendor and elegance in the conventionally dry sciences of math and electronics. They saw programming as a form of artistic expression and the computer as an instrument of that art. Their desire to dissect and understand wasn't intended to demystify artistic endeavors; it was simply a way to achieve a greater appreciation of them. These knowledge-driven values would eventually be called the *Hacker Ethic*: the appreciation of logic as an art form and the promotion of the free flow of information, surmounting conventional boundaries and restrictions for the simple goal of

better understanding the world. This is not a new cultural trend; the Pythagoreans in ancient Greece had a similar ethic and subculture, despite not owning computers. They saw beauty in mathematics and discovered many core concepts in geometry. That thirst for knowledge and its beneficial by-products would continue on through history, from the Pythagoreans to Ada Lovelace to Alan Turing to the hackers of the MIT model railroad club. Modern hackers like Richard Stallman and Steve Wozniak have continued the hacking legacy, bringing us modern operating systems, programming languages, personal computers, and many other technologies that we use every day.

How does one distinguish between the good hackers who bring us the wonders of technological advancement and the evil hackers who steal our credit card numbers? The term *cracker* was coined to distinguish evil hackers from the good ones. Journalists were told that crackers were supposed to be the bad guys, while hackers were the good guys. Hackers stayed true to the Hacker Ethic, while crackers were only interested in breaking the law and making a quick buck. Crackers were considered to be much less talented than the elite hackers, as they simply made use of hacker-written tools and scripts without understanding how they worked. *Cracker* was meant to be the catch-all label for anyone doing anything unscrupulous with a computer—pirating software, defacing websites, and worst of all, not understanding what they were doing. But very few people use this term today.

The term's lack of popularity might be due to its confusing etymology—*cracker* originally described those who crack software copyrights and reverse engineer copy-protection schemes. Its current unpopularity might simply result from its two ambiguous new definitions: a group of people who engage in illegal activity with computers or people who are relatively unskilled hackers. Few technology journalists feel compelled to use terms that most of their readers are unfamiliar with. In contrast, most people are aware of the mystery and skill associated with the term *hacker*, so for a journalist, the decision to use the term *hacker* is easy. Similarly, the term *script kiddie* is sometimes used to refer to crackers, but it just doesn't have the same zing as the shadowy *hacker*. There are some who will still argue that there is a distinct line between hackers and crackers, but I believe that anyone who has the hacker spirit is a hacker, despite any laws he or she may break.

The current laws restricting cryptography and cryptographic research further blur the line between hackers and crackers. In 2001, Professor Edward Felten and his research team from Princeton University were about to publish a paper that discussed the weaknesses of various digital watermarking schemes. This paper responded to a challenge issued by the Secure Digital Music Initiative (SDMI) in the SDMI Public Challenge, which encouraged the public to attempt to break these watermarking schemes. Before Felten and his team could publish the paper, though, they were threatened by both the SDMI Foundation and the Recording Industry Association of America (RIAA). The Digital Millennium Copyright Act (DCMA) of 1998 makes it illegal to discuss or provide technology that might be used to bypass industry consumer controls. This same law was used against Dmitry Sklyarov, a Russian computer programmer and hacker. He had written software to circumvent

overly simplistic encryption in Adobe software and presented his findings at a hacker convention in the United States. The FBI swooped in and arrested him, leading to a lengthy legal battle. Under the law, the complexity of the industry consumer controls doesn't matter—it would be technically illegal to reverse engineer or even discuss Pig Latin if it were used as an industry consumer control. Who are the hackers and who are the crackers now? When laws seem to interfere with free speech, do the good guys who speak their minds suddenly become bad? I believe that the spirit of the hacker transcends governmental laws, as opposed to being defined by them.

The sciences of nuclear physics and biochemistry can be used to kill, yet they also provide us with significant scientific advancement and modern medicine. There's nothing good or bad about knowledge itself; morality lies in the application of knowledge. Even if we wanted to, we couldn't suppress the knowledge of how to convert matter into energy or stop the continued technological progress of society. In the same way, the hacker spirit can never be stopped, nor can it be easily categorized or dissected. Hackers will constantly be pushing the limits of knowledge and acceptable behavior, forcing us to explore further and further.

Part of this drive results in an ultimately beneficial co-evolution of security through competition between attacking hackers and defending hackers. Just as the speedy gazelle adapted from being chased by the cheetah, and the cheetah became even faster from chasing the gazelle, the competition between hackers provides computer users with better and stronger security, as well as more complex and sophisticated attack techniques. The introduction and progression of intrusion detection systems (IDSs) is a prime example of this co-evolutionary process. The defending hackers create IDSs to add to their arsenal, while the attacking hackers develop IDS-evasion techniques, which are eventually compensated for in bigger and better IDS products. The net result of this interaction is positive, as it produces smarter people, improved security, more stable software, inventive problem-solving techniques, and even a new economy.

The intent of this book is to teach you about the true spirit of hacking. We will look at various hacker techniques, from the past to the present, dissecting them to learn how and why they work. Included with this book is a bootable LiveCD containing all the source code used herein as well as a preconfigured Linux environment. Exploration and innovation are critical to the art of hacking, so this CD will let you follow along and experiment on your own. The only requirement is an *x*86 processor, which is used by all Microsoft Windows machines and the newer Macintosh computers—just insert the CD and reboot. This alternate Linux environment will not disturb your existing OS, so when you're done, just reboot again and remove the CD. This way, you will gain a hands-on understanding and appreciation for hacking that may inspire you to improve upon existing techniques or even to invent new ones. Hopefully, this book will stimulate the curious hacker nature in you and prompt you to contribute to the art of hacking in some way, regardless of which side of the fence you choose to be on.