

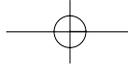
# Introduction

Hardware hacking. Mods. Tweaks. Though the terminology is new, the concepts are not: A gearhead in the 1950s adding a custom paint job and turbo-charged engine to his Chevy Fleetline, a '70s teen converting his ordinary bedroom into a “disco palace of love,” complete with strobe lights and a high-fidelity eight-track system, or a techno-geek today customizing his computer case to add fluorescent lighting and slick artwork. Taking an ordinary piece of equipment and turning it into a personal work of art. Building on an existing idea to create something better. These types of self-expression can be found throughout recorded history.

When Syngress approached me to write this book, I knew they had hit the nail on the head. Where else could a geek like me become an artistic genius? Combining technology with creativity and a little bit of skill opened up the doors to a whole new world: hardware hacking.

But why do we do it? The reasons might be different for all of us, but the end result is usually the same. We end up with a unique thing that we can call our own—imagined in our minds and crafted through hours, days, or years of effort. *And* doing it on our own terms.

Hardware hacking today has hit the mainstream market like never before. Computer stores sell accessories to customize your desktop PC. Web sites are popping up like unemployed stock brokers to show off the latest hacks. Just about any piece of hardware can serve as a candidate to be hacked. Creativity and determination can get you much farther than most product developers could ever imagine. Hardware hacking is usually an individual effort, like creating a piece of art.



## xxxvi Introduction

However, just like artists, hackers sometimes collaborate and form communities of folks working toward a similar goal.

The use of the term *hacker* is a double-edged sword and often carries a mythical feel. Contrary to the way major media outlets enjoy using the word to describe criminals breaking into computer systems, a hacker can simply be defined as somebody involved in the exploration of technology. And a *hack* in the technology world usually defines a new and novel creation or method of solving a problem, typically in an unorthodox fashion.

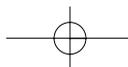
The philosophy of most hardware hackers is straightforward:

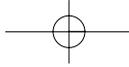
- Do something with a piece of hardware that has never been done before.
- Create something extraordinary.
- Harm nobody in the process.

Hardware hacking arguably dates back almost 200 years. Charles Babbage created his difference engine in the early 1800s—a mechanical form of hardware hacking. William Crookes discovered the electron in the mid-1800s—possibly the first form of electronics-related hardware hacking. Throughout the development of wireless telegraphy, vacuum tubes, radio, television, and transistors, there have been hardware hackers—Benjamin Franklin, Thomas Edison, and Alexander Graham Bell, to name a few. As the newest computers of the mid-20<sup>th</sup> century were developed, the ENIAC, UNIVAC, and IBM mainframes, people from those academic institutions fortunate enough to have the hardware came out in droves to experiment. With the development and release of the first microprocessor (Intel 4004) in November 1971, the general public finally got a taste of computing. The potential for hardware hacking has grown tremendously in the past decade as computers and technology have become more intertwined with the mainstream and everyday living.

Hardware hacks can be classified into four different categories, though sometimes a hack falls into more than one:

1. **Personalization and customization** Think “hot rodding for geeks,” the most prevalent of hardware hacking. This includes things such as case modifications, custom skins and ring tones, and art projects like creating an aquarium out of a vintage computer.
2. **Adding functionality** Making the system or product do something it wasn’t intended to do. This includes things such as converting the





iPod to run Linux, turning a stock iOpener into a full-fledged PC, or modifying the Atari 2600 to support stereo sound and composite video output.

3. **Capacity or performance increase** Enhancing or otherwise upgrading a product. This includes things such as adding memory to your favorite personal digital assistant (PDA), modifying your wireless network card to support an external antenna, or overclocking your PC's motherboard.
4. **Defeating protection and security mechanisms** This includes things such as removing the unique identifier from CueCat barcode scanners, finding Easter eggs and hidden menus in a TiVo or DVD player, or creating a custom cable to unlock the secrets of your cell phone. Theft-of-service hacks fall into this category, but this book doesn't cover them.

Creating your own hardware hacks and product modifications requires at least a basic knowledge of hacking techniques, reverse-engineering skills, and a background in electronics and coding. All the information you'll need is in the pages of this book. And if a topic isn't covered in intimate detail, we include references to materials that do. If you just want to do the hack without worrying about the underlying theory behind it, you can do that, too. The step-by-step sections throughout each chapter include pictures and "how to" instructions. The details are in separate sections that you can skip right over and get to the fun part—voiding your warranty!

This book has something for everyone from the beginner hobbyist with little to no electronics or coding experience to the self-proclaimed "gadget geek" and advanced technologist. It is one of the first books to bring hardware hacking to the mainstream. It is meant to be fun and will demystify many of the hacks you have seen and heard about. We, all the contributors to this project, hope you enjoy reading this book and that you find the hacks as exciting and satisfying as we have.

If your friends say "Damn, now *that's* cool," then you know you've done it right.

—Joe Grand, the hardware hacker formerly known as Kingpin  
January 2004

