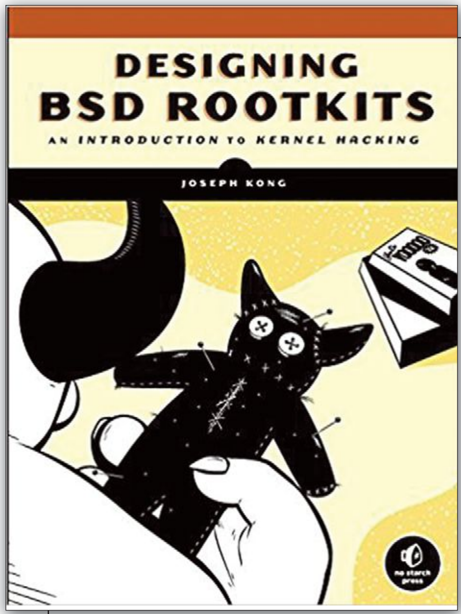
BOOKreview by Steven Kreuzer



Designing BSD Rootkits:
An Introduction to Kernel Hacking
by Joseph Kong

Publisher	*No Starch Press (2007)*
Print List Price	*$29.95*
Digital List Price	*$23.95*
ISBN-10	*1593271425*
ISBN-13	*978-1593271428*
Pages	*142 pages*

It's hard to believe that it was 10 years ago when I stumbled across *Designing BSD Rootkits*, by Joseph Kong, as I was wandering around a local bookstore. Not only was this a book on a technical topic that I happen to find pretty fascinating, but it also used FreeBSD as the reference platform! Since the focus of this issue is security, I thought it would be fun to dust off my copy and take another look.

At this point you might be asking what exactly is a rootkit? Once an attacker gains unauthorized access to your system, the goal is usually to avoid detection and maintain control of the machine for as long as possible. This is usually accomplished by deploying a rootkit, which is a collection of utilities designed to hide the intrusion as well as to maintain privileged access. While this book focuses on the FreeBSD operating system, the concepts that are covered can be applied to other operating systems as well. Even if you do not work with FreeBSD on a day-to-day basis, you will still gain valuable insights into computer security that can be applied to pretty much any system.

One of the things I love about this book is that it cuts right to the chase. This first chapter immediately dives into the topic of loadable kernel modules, which is the double-edge sword that allows your system administrator to add functionality to your running system without having to reboot, but is used by an attacker to load malicious code that can help cover their tracks. Kong wastes no time, because on the fourth page of this book you are presented with a bare bones kernel module which you can build, load, and unload on your machine. Following a tutorial style format, each chapter slowly builds upon the previous one, introducing more advance functionality. By the end of the book your simple kernel module which just printed 'Hello, world!' will be a complete and functional rootkit capable of evading host-based intrusion detection systems such as tripwire.

In the final chapter, the focus shifts from offense to defense with a chapter dedicated to the detection of rootkits. Amusingly, this is also the second shortest chapter with very few pieces of example code. The overall theme is that once a rootkit is in place, detection isn't easy. Software designed to detect a rootkit can just as easily be subverted by the very rootkit it is looking for, and the kernel is a very big place that offers lots of places to hide. However, since you are now armed with a much better understanding of what is actually going on under the hood, you will be in a superior position should you ever have to do battle with a rootkit.

While the book is thin, weighing in at only 142 pages, it manages to pack in a wealth of information and example code. To get the most out of this book, it will be helpful to have some understanding of both C and assembly as well as some knowledge of FreeBSD kernel internals. However, I found Kong's tutorial-style approach combined with the very well documented code examples made the concepts incredibly accessible. The fact that each chapter provided at least one real-world application for each topic that he covers makes it even easier to experiment and build upon the examples on your own. You could read the book from cover to cover in a few hours, but I would highly recommend that you sit yourself down in front of your computer with a copy of this by your side while you play around with the provided examples. It becomes quite easy to quickly lose a few hours as you load and unload a kernel module designed for nefarious purposes.

Don’t let the fact that the book was published a decade ago scare you away. Not only does this book serve as a great introduction to FreeBSD kernel module programming, the methods and tactics that are covered are still in use today. One thing to note is that a lot has changed in FreeBSD, and when this book was first written, the author was using FreeBSD 6.0-STABLE on a 32-bit platform. Not all the code examples may properly compile on a newer system, so it becomes an exercise for the reader to debug and update the provided examples. With all that said, I highly recommend you add a copy to your bookshelf. ●

STEVEN KREUZER is a FreeBSD Developer and Unix Systems Administrator with an interest in retro-computing and air-cooled Volkswagens. He lives in Queens, New York, with his wife, two daughters, and dog.