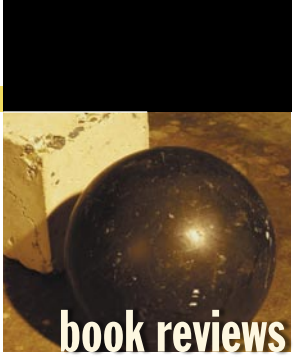
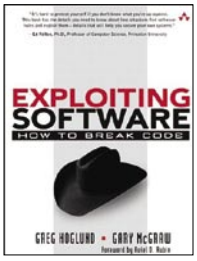


Exploiting Software: How to Break Code
Greg Hoglund, Gary McGraw
Addison-Wesley, 2004, $49.99, ISBN: 0201786958



A book with a dark-gray hat on its cover and the subtitle "How to Break Code" makes a strong statement. It does not disappoint. It covers many form of exploiting software that you never dared to explore.

The book approaches its problem from many security disciplines. It takes on the reverse-engineering angle to break copyright protection systems or to find software defects. It takes the pentest (penetration test) view when it explores how to attack server-side software, with local and remote attack options. It describes the botnet (robot network) master's options when it targets client software problems. It shows how to hide malware (malicious software) via the rootkit approach, diving deep—even into flash memory—and evading forensic analysis.

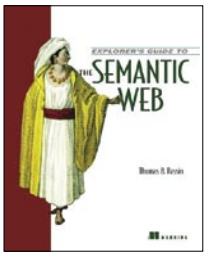
The authors also present more conceptual views, such as the root cause of software security problems, 49 attack patterns, how to craft malicious input, and buffer overflows in all variations. Each topic includes a tutorial, sample systems or code, and known exploits using these techniques. If the topic is unfamiliar, the tutorial may be insufficient, but links to further information are provided. The sample code is clear enough to allow smarter scripters to elaborate on it. There are not many details on the known exploits, but a simple Web search on any of the key terms will provide all that are necessary.

The book is about 450 pages, and contains eight chapters. The three longest chapters are on reverse engineering, buffer overflow, and rootkits. The others are on software, attack patterns, exploiting server software, and exploiting client software

Who should read this book? The authors start by defending why anyone would *write* such a book. They show that anything they describe has been exploited already. They spell out how it was done, loud and clear. This takes away ignorance. So, if your job is to build secure software systems, to implement license or copyright protection systems, to pentest systems, or to do forensic analysis, you will benefit from reading the book.

Ed Felten, Princeton University professor of computer science, is quoted on the cover: "It's hard to protect yourself if you don't know what you're up against." But having that knowledge, after reading this book, may not improve your peace of mind. —*A. Mariën*

Explorer's Guide to the Semantic Web
Thomas B. Passin
Manning Publications, 2004, $39.95, ISBN: 1932394206



This book was written by Thomas Passin, a principal systems engineer specializing in data modeling, Web databases, and XML projects. Thus, it is not unexpected to find that he covers semantic Web issues from a database and data modeling perspective, especially the chapter devoted to RDF (resource description framework) and topic maps.

I might summarize this book as a gentle, semi-informal introduction to the semantic Web. It starts with a brief explanation of what the semantic Web consists of and the technologies involved.

Chapter 2 is an intuitive introduction to RDF. Chapter 3 presents topic maps, which could be thought of as a graph-based data model for representing lists, glossaries, thesauri, tables of contents, and indices.

Chapter 4 is devoted to the semantic annotation of Web resources, and Chapter 5 addresses searching with the aid of semantic Web technologies, such as annotation and self-describing meta-data, and social analysis.

Chapters 6 and 7 include brief introductions to logic, ontologies, and ontology languages such as OWL (Web ontology language) and RDF Schema. Chapter 8 introduces semantic Web services, and Chapter 9 looks at the field of intelligent agents.

The last two chapters are devoted to distributed trust and belief (the use of cryptography, keys, digital signing, and data provenance), and the future of the semantic Web. The book also includes an appendix of case studies.

I recommend this book to students, developers, and researchers who are curious about the semantic Web or who are looking for an upper-level viewpoint that just covers the main infrastructure and technologies involved.

This book is easy to read, but it is lacking in deep insights on complex issues such as ontology management, semantic Web services, description logics, and semantic grids. —*Carlos F. Enguix*