

## Book Review: A Bug Hunter's Diary

Review by Tristan Lawson, CISSP, MCSE: Security, GCIH, OSCP et al

So often as security professionals we hear how bug hunters both black hat and white hat find vulnerabilities and release them to the vendor or use them for monetary gain. We wonder how they actually went about finding these vulnerabilities and what hurdles they had to jump to find them. "A Bug Hunter's Diary: A Guided Tour Through the Wilds of Software Security" by Tobias Klein focuses on helping different levels of security professionals understand the approaches used to uncover vulnerabilities, testing the vulnerabilities found and finally reporting on those vulnerabilities. It is short and to the point and offers nothing but valuable content with little to no fluff content.

The book was written as though Tobias was writing in a journal as he was progressing through his research of a particular application. Each chapter is a separate journal entry focused on a single application into which he dug and eventually found a vulnerability. He then determined if it was exploitable and in turn released it to either the vendor or to a vulnerability broker. This is a fascinating look into the heart of a sector of the security economy not previously exposed to a wider audience.

After the break, look for a link to a free download of Chapter 2: "Back to the 90s"

Discuss in Forums {mos\_smf\_discuss:Book Reviews}

[Click Here to Download Chapter 2: "Back to the 90s"](#)

"A Bug Hunter's Diary" starts out by introducing the reader to the idea of how to look for vulnerabilities in software, and it then progresses to how exploiting these vulnerabilities, one can gain control of the operating system. The book then progresses through seven different journal entries ranging from media players, system kernels, both standard and browser-based applications and smart phones.

Those who read this book do not need to be programmers in any language, though it certainly helps if you understand C and assembly, since most of the code segments is in one of those languages. Tobias goes to great lengths to explain every line which is important in an assembly dump or code segment, so even someone who is not well versed in programming or these particular languages can still get a good feel for what is going on and understand the flow of the application and the vulnerability discovery process.

If you are a more visual person, such as myself, then Tobias details the flow of the applications and how it is exploited utilizing very useful diagrams and call graphs. The diagrams demonstrate how things communicate to the OS, and, for instance in Solaris, what a message block stream communication from user to kernel space would look like logically.

Programmers will gain many benefits from "A Bug Hunter's Diary." So often software developers may even know what is best practice, and, just like most general users, they get lazy or busy. In turn, they do not execute best practices while coding, never observing the effects of these actions and experiencing what possible byproducts could result. This is illustrated in Chapter 2 "Back to the 90s," where a patch was released for VLC to enforce a fixed size to a user supplied value, but due to an oversight a variable of type signed int allowed the vulnerability to still work. These real-world scenarios show the consequences of being careless by doing something as simple as not validating user input.

Tobias writes out a 'Lessons Learned' section at the end of each chapter with great tips for programmers to help them understand what they should avoid in programming and why. Programmers should read this book purely for the insight gained into how researchers and attackers seek out the shortcuts programmers have taken, how to go about exploiting them and finally exposing the consequences of such actions.

Not all vulnerabilities are a result of poor coding and programmer shortcuts. Tobias touches upon vulnerabilities that are created by the compiler during the compilation of the application and how two functions written separately can open the door for potential abuse when put together. It is in these instances where Tobias demonstrates the importance of manual code testing since automated code review can easily miss these.

The book also discusses and demonstrates the patching process with software maintainers during a coordinated vulnerability disclosure showing the actual code versus the patched code, walking through the changes which were made, and explaining why it is fixed after the patch.

Readers will find amusement (as I did) in the time lines from vendor disclosure of a vulnerability to the actual release of the patch. Some vendors take way too long to release a patch, sometimes resulting in other researchers finding the same vulnerability, and, instead of contacting the vendor, they release it to the public instead resulting in quite a mess.

Tobias discusses what interests him and why certain exploits work. He also digs deeper into certain areas such as why a particular exploit worked when security cookies, DEP and ASLR were available on the OS, when, logically with these security mechanisms in place, the exploit should not have worked against the discovered vulnerability.

The book dives into further detail on reverse engineering techniques used to discover why applications may be set to do something like not Optin for DEP protection. Tools and techniques are given a once over to show how you discover these details in the software and the thought process is written out eloquently.

The title and the subtitle say it all "A Guided Tour Through the Wilds of Software Security." This book is a focused glimpse into the underpinnings of vulnerability research and exploitation and the approaches described. This topic is a breath of fresh air, since it is not often exposed how researchers go through the process of finding bugs and what to do if one is found.

"A Bug Hunter's Diary" by Tobias Klein is a book that should grab the interest of just about any information security professional, be it new to the field or a veteran. If you are a developer, you will learn methods to code securely and avoid the pitfalls demonstrated in the book. If you are an information security enthusiast, you will read of the many ways software can be exploited. Finally if you are a penetration tester, you will appreciate the countless ways software vulnerabilities can be discovered and exploited. Whatever your background, Tobias Klein's book offers something to those who have an interest in software security. And the fact that it is presented in such an approachable format, makes it easy to dive into the life of a bug hunter.

#### Additional Info & Resources

Please be sure to visit the companion website of the book:

<http://www.trapkit.de/books/bhd/en.html>

Tristan Lawson, CISSP, OSCP, C|EH, E|CSA, C|HFI, GXPn, GWAPT, GCIH, GISP, GSEC, MCITP:EA, CCNA, FCT, FCNSP, JNCIA, JNCIA-FWV, MCSE Security, A+, Net+, Server+, Security+

Tristan Lawson is a Senior Information Security Consultant for Infogressive Inc. out of Lincoln, NE. He brings 10 years' experience in Information technology and is able to exercise his breadth of knowledge performing penetration testing, network security assessments, and assisting companies in deploying security technology. In addition to his passion for Information Security, Tristan also has a passion for marine biology and is actively pursuing a Master's degree in Information Assurance from University Nebraska Omaha.

