“Give a man an exploit and you make him a hacker for a day; teach a man to exploit bugs and you make him a hacker for a lifetime.” — Felix “FX” Lindner

Seemingly simple bugs can have drastic consequences, allowing attackers to compromise systems, escalate local privileges, and otherwise wreak havoc on a system.

*A Bug Hunter’s Diary* follows security expert Tobias Klein as he tracks down and exploits bugs in some of the world’s most popular software, like Apple’s iOS, the VLC media player, web browsers, and even the Mac OS X kernel. In this one-of-a-kind account, you’ll see how the developers responsible for these flaws patched the bugs—or failed to respond to them at all.

Along the way you’ll learn how to:

- Use field-tested techniques to find bugs, like identifying and tracing user input data and reverse engineering
- Exploit vulnerabilities like NULL pointer dereferences, buffer overflows, and type conversion flaws
- Develop proof-of-concept code that verifies the security flaw
- Report bugs to vendors or third-party brokers

*A Bug Hunter’s Diary* is packed with real-world examples of vulnerable code and the custom programs used to find and test bugs. Whether you’re hunting bugs for fun, for profit, or to make the world a safer place, you’ll learn valuable new skills by watching over the shoulder of a professional bug hunter in action.

**ABOUT THE AUTHOR**

Tobias Klein is a security researcher and founder of NESO Security Labs, an information security consulting and research company. He is the author of two information security books published in the German language by dpunkt.verlag.
INDEX

Numbers
4.4BSD, 130
4X movie file format, 53

A
AAC (Advanced Audio Coding), 136
ActiveX, 71
Address Space Layout Randomization (ASLR), 19–21, 179–182
Advanced Audio Coding (AAC), 136
ALWIL Software, 87
antivirus products, 87
Apache webserver, 137
Apple
  GNU Debugger version, 173
  iPhone, 133
  MacBook, 113
ARM CPU, 7, 140, 146
assembly syntax
  AT&T, 124, 173
  Intel, 93, 140, 173
ASLR (Address Space Layout Randomization) 19–21, 179–182
Audio Toolbox (Apple iOS audio framework), 134
avast! antivirus product, 87

B
Blue Screen of Death (BSoD), 109
brute force technique, 63, 125
BSoD (Blue Screen of Death), 109
buffer overflows, 5, 9, 81, 142, 149, 180, 183
bug hunting, definition of, 3

C
Celestial (Apple iOS audio framework), 134
checksec.sh, 183–184
Cisco, 71, 84
Common Vulnerabilities and Exposures Identifiers (CVE-IDs), 23
CVE-2007-4686, 130
CVE-2008-568, 49
CVE-2008-1625, 110
CVE-2008-3558, 84
CVE-2008-4654, 22
CVE-2009-0385, 69
CVE-2010-0036, 147
COMRaider, 72
coordinated disclosure, 18
Core Audio (Apple iOS audio framework), 134
cross-site scripting (XSS), 75
CTL_CODE, 97
CurrentStackLocation, 95
CVE-IDs. See Common Vulnerabilities and Exposures Identifiers
Cygwin environment, 21

D
Data Execution Prevention (DEP), 19–21, 179–182
data transfer type, 97
debuggers, 6
   The GNU Debugger (gdb), 7, 121, 140, 171–176
   Immunity Debugger, 7, 16
   The Modular Debugger (mdb), 7, 37, 163–165
   OllyDbg, 7
   WinDbg, 7, 76–77, 92–95, 99, 107, 165–170
demuxer, 10, 52
dep (Data Execution Prevention), 19–21, 179–182
DeviceIoControl(), 90
Direct Kernel Object Manipulation (DKOM), 110
disassemblers, 7
DispCallFunc(), 76
DKOM (Direct Kernel Object Manipulation), 110
double frees, 6
DRIVER_OBJECT, 90
DriverView, 88
dynamic analysis, 4

E
ELF (Executable and Linkable Format), 61, 157
Enhanced Mitigation Experience Toolkit (EMET), 22
Executable and Linkable Format (ELF), 61, 157
exploit, 3
   for avast! antivirus product vulnerability, 110
development of, 8
   for FFmpeg vulnerability, 65
   for Mac OS X kernel vulnerability, 129
   for Sun Solaris kernel vulnerability, 48
   for VLC media player vulnerability, 18
   for WebEx vulnerability, 83

F
FFmpeg multimedia library, 51, 155
FreeBSD, 130
full disclosure, 18, 84
fuzzing, 4, 134

G
gdb (The GNU Debugger), 7, 121, 140, 171–176
Global Offset Table (GOT), 61, 67, 157, 183
GNU Debugger, The (gdb), 7, 121, 140, 171–176
GOT overwrite, 67, 157–161
/GS, 19, 152, 179–182

H
heap buffer overflows, 149. See also buffer overflows
heap-memory management, 6
heap mitigation techniques, 179
heap spraying techniques, 83, 129

I
IDA Pro (Interactive Disassembler Professional), 7, 78, 88, 181
Immunity Debugger, 7, 16
input/output controls (IOCTL), 26, 88, 113
ioctl(), 115
instruction alignment, 146
instruction pointer, 7, 150
Intel, 7, 149
Interactive Disassembler Professional (IDA Pro), 7, 78, 88, 181
Internet Explorer, 71
IoCreateDevice(), 88
IOCTL (input/output controls), 26, 88, 113
ioctl(), 115
I/O request packet (IRP), 95
_IO_STACK_LOCATION, 96
iPhone, 133
IRP (I/O request packet), 95
IRP_MJ_DEVICE_CONTROL, 90
jmp reg technique, 18, 19
kernel debugging, 7, 37, 88, 121, 167, 173
Kernel Debug Kit, 174
kernel driver, 87
kernel panic, 32, 37–38, 120, 165
kernel space, 39, 102
KeSetEvent(), 107
Linux
Debian, 157, 183
debugging the Mac OS X kernel with, 121, 173
and exploit mitigation techniques, 180, 183
fuzzing the iPhone with, 134
gdb, debugger for, 7
Red Hat, 173
stack buffer overflows under, 151
Ubuntu, 56, 63, 151
little-endian, 17, 143
LookingGlass, 21
Mac OS X, 7, 113, 173
mdb (The Modular Debugger), 7, 37, 163–165
mediaserverd, 134
memcpy(), 101, 142
memory corruption, 6, 140, 149, 157
memory errors, 6
memory leak, 129, 140
METHOD_BUFFERED, 99
MindshaRE, 76
mmmap(), 44
MobileSafari, 133
Modular Debugger, The (mdb), 7, 37, 163–165
Most Significant Bit (MSB), 156
movie header atom, 144
movsx, 5
MSB (Most Significant Bit), 156
non-maskable interrupt (NMI), 122
NULL pointer dereference, 6, 32, 51, 153–154
objdump, 63, 161, 184
OS X, 7, 113, 173
parser, 9
PLT (Procedure Linkage Table), 158–160
privilege escalation, 110, 129
Procedure Linkage Table (PLT), 158–160
program counter, 7, 150
Python, 74
QuickTime (File Format Specification), 144
readelf, 161
RELRO, 67–69, 183–186
rep movsd, 101
responsible disclosure, 18
return address (RET), 150
runtime link editor (rtld), 157, 159

saved frame pointer (SFP), 150–151
security advisories
   TKADV2007-001, 131
   TKADV2008-002, 111
   TKADV2008-009, 85
   TKADV2008-010, 24
   TKADV2008-015, 50
   TKADV2009-004, 70
   TKADV2010-002, 148
security cookie, 19, 152, 179-182
SFP (saved frame pointer), 150–151
sign bit, 156
sign-extension vulnerabilities, 5
SiteLock, 84
Solaris
   kernel, 25
   mdb, debugger for, 7
Solaris Zones, 39, 186-189
sprintf(), 80
stack buffer overflows, 149. See also buffer overflows
stack canary, 151, 180
stack frame, 150
static analysis, 4
STREAMS, 27

Tipping Point, Zero Day Initiative (ZDI), 18
TiVo file format, 10
type conversion, 51, 117, 154

uninitialized variables, 6
user space, 27, 39, 51, 90, 129

VBScript, 74
VCP (Vulnerability Contributor Program), 18, 84
Verisign iDefense Labs, Vulnerability Contributor Program (VCP), 18, 84
VideoLAN, 9
VirusTotal, 87
VLC media player, 9, 51, 65
VMware, 88, 167–170
vulnerability brokers, 18
   Tipping Point, 18
   Verisign iDefense Labs, 18, 84
Vulnerability Contribution Program (VCP), 18, 84
vulnerability rediscovery, 84

WebEx Meeting Manager, 71
WinDbg, 7, 76–77, 92–95, 99, 107, 165–170
Windows I/O manager, 95
Windows Vista, 10, 19, 152, 156, 181
Windows XP, 71, 88, 107, 167, 180
WinObj, 90

XNU kernel, 113, 174
XSS (cross-site scripting), 75
xxd, 136

Zero Day Initiative (ZDI), 18
zero page, 39–46, 153