

# INDEX

## A

- access time of file, preventing change, 107–114
- allproc list, 42–43
- application programming interfaces (APIs)
  - HIDS software and, 92
  - hooking to alter results, 24
  - problems from trusting, 123
- arg parameter, in SYSCALL\_MODULE, 9
- argument structure, padding in, 7

## B

- behavior, rootkit detection by, 119
- bsd.kmod.mk, 4–5

## C

- call hooking. *See* hooking
- call statement
  - patching, 70–73
  - in x86 assembly, 70
- cdevp\_list queue, 60
- cdevsw structure, 14–15
- char\* filename field, in struct linker\_file, 103
- char p\_comm[MAXCOMLEN + 1]; field, in proc structure, 42
- character device, 14
  - hooking, 59–62
- character device drivers, entry points, 15
- character device modules, 14–21
  - cdevsw structure, 14–15
  - character device functions, 15–16
  - device registration routine, 16
  - example, 17–19
  - testing character device, 19–21
- character device switch table, 59
  - modifying, 60–62

- chdir system call, 30
- chmod system call, 30
- chown system call, 30
- code byte patches, finding, 125
- communication protocols, 30–32
  - hooking, 32–34
- compiling executable file, 5
- copyin function, 13
- copyinstr function, 13
- copyout function, 13
- copystr function, 13
- corefile parameter, for kvm\_openfiles function, 64
- curthread, 123

## D

- data corruption, 56–57
  - symmetric multiprocessing (SMP) and, 39
- data parameter, for DECLARE\_MODULE macro, 3
- d\_close entry point, for character device driver, 15
- ddb() [kernel-mode debugger], 81
- debug message, output from mkdir system call, 24
- Debugging Kernel Problems* (Lehey), 22
- DECLARE\_MODULE macro, 3
- destroy\_dev function, 16
- /dev/kmem device, 63
- device driver. *See* KLD (Dynamic Kernel Linker)
- DEV\_MODULE macro, 19
- devmtx mutex, 60
- d\_ioctl1 entry point, for character device driver, 15
- DIP\_SET macro, 111
- Direct Kernel Object Manipulation (DKOM), 37–57

- detection, 123–125
- hiding running process, 41–46
  - allproc list, 42–43
  - example, 43–46
  - further steps, 46–51
  - proc structure, 41–42
- hiding with, 51
- kernel queue data structures, 37–39
- synchronization issues, 39–41
- dirent structure, 98
- DKOM. *See* Direct Kernel Object Manipulation (DKOM)
- d\_open entry point, for character device driver, 15
- doubly-linked list, 38
- d\_poll entry point, for character device driver, 15
- d\_read entry point, for character device driver, 15
- d\_write entry point, for character device driver, 15
- Dynamic Kernel Linker (KLD) Facility, 1

## E

- enum p\_state; field, in proc structure, 42
- errbuf parameter, for kvm\_openfiles function, 64
- evh parameter, in SYSCALL\_MODULE, 9
- exclusive lock, 40–41
- execfile parameter, for kvm\_openfiles function, 64
- executable file, compiling, 5
- executing system call, 11
  - without C code, 12
- execution redirection, 92–95
- execve system call, 30
  - hooking, 92–95
- \_exit() system call function, 51

## F

- files
  - displaying status of those dynamically linked into kernel, 21–22
  - hiding, 96–101
  - preventing access, modification, and change time updates, 107–114
  - example, 112–114
- finding
  - hidden ports, 125
  - hidden processes, 123–124
  - hooks, 120–123
  - inline function hooks, 125

- flags parameter
  - for kvm\_openfiles function, 64
  - for malloc function, 73–74
- free function, 74
- FREE macro, 74–75
- FreeBSD, xvi
  - setting up machine, 22
  - sysent[] as system call table, 7
  - version 3.0 changes to kernel module subsystem, 1
  - virtual memory parts, 6*n*
- The FreeBSD Developer's Handbook*, 22
- <fs/devfs/devfs\_int.h> header, 60
- functions, finding with patched code, 125

## G

- getdirent system call, 30
  - hooking, 96–101
- Giant lock, 102

## H

- hard-coded offsets, avoiding use of, 83
- hash table, 47*n*
- hashinit function, 47
- "Hello, world!" module, 4–5
- hidden ports, finding, 125
- hidden processes, finding, 123–124
- hiding
  - KLD (Dynamic Kernel Linker), 101–107
  - open TCP-based port, 52–56
- hooking, 23–35
  - character device, 59–62
  - common system call, 29–30
  - communication protocols, 32–34
  - detection, 120–123
  - getdirent system call, 96–101
  - inline function, 81–88
    - finding, 125
  - kernel process tracing, 28–29
  - keystroke logging, 26–28
  - system call, 24–26
- Host-based Intrusion Detection Systems (HIDSes)
  - bypassing, 92
  - purpose of, 91
  - rootkit to bypass, 91–117
    - execution redirection, 92–95
    - file hiding, 96–101

- hiding KLD, 101–107
  - Tripwire, avoiding recognition by, 114–116
- hot patching, 90

## I

- ICMP (Internet Control Message Protocol), 32–34
  - icmp\_input\_hook function, 32–34
  - inetsw[] switch table, 31–32
  - inline function, hooking, 81–88
    - finding, 125
  - inpcb structure, 52–53
  - inpcbinfo structure, 53
    - removing from tcbinfo.listhead list, 54–56
  - int\_p\_flag; field, in proc structure, 41
  - int\_refs; field, in struct linker\_file, 102
  - Internet Control Message Protocol (ICMP), 32–34
  - Internet protocol control block, 52
  - ioctl system call, 30

## J

- jump, unconditional, 81

## K

- kdump() utility, 28–29
- kernel
  - corrupting data, 56–57
  - detecting memory patching, 125
  - KLD registration with, 3
  - memory allocation, 73–77
    - from user space, 77–81
  - queue data structures, 37–39
    - synchronization, 39–41
  - running
    - loading and unloading code into, 5
    - userland code to patch, 63–90
  - virtual memory
    - interface for accessing, 63
    - patching code bytes, 66–70
- Kernel Data Access Library (libkvm), 63–66
- kernel-mode debugger, 81
- kernel module
  - function to return status, 10
  - modid for, 10
  - structure, 103–104
- Kernel Object Hooking (KOH), hooking character device, 59–62

- kernel panic, 13*n*, 56, 77, 88
  - avoiding, 44
- kernel process tracing, 28–29
- kernel source tree, 22
- kernel space, 6*n*
  - functions for data manipulation in user space, 12–13
- keystroke logging, with system call hook, 26–28
- kill system call, 30
- KLD (Dynamic Kernel Linker), 1
  - "Hello, world!" module, 4–5
  - hiding, 101–107
  - initialization and shutdown routines for, 2–3
  - registration with kernel, 3
- kldload system call, 5, 30
- kldstat() command, 21, 101
- kldunload system call, 5, 30
- ktrace() utility, 28–29
- kvm\_close function, 66
- kvm\_geterr function, 65
- kvm\_nlist function, 64–65
- kvm\_openfiles function, 64
- kvm\_read function, 65
- kvm\_write function, 65

## L

- libkvm (Kernel Data Access Library), 63–66
- linesw[] switch table, 35
- linker files, 21–22
  - KLD structure in, 101
- linker\_file structure, 102–103
- linker\_files list, 102
- LIST\_ENTRY macro, 38–39
- LIST\_ENTRY(inpcb) inp\_list; field, in inpcb structure, 52
- LIST\_ENTRY(proc) p\_hash; field, in proc structure, 42
- LIST\_ENTRY(proc) p\_list; field, in proc structure, 41
- LIST\_FOREACH macro, 39
- LIST\_HEAD macro, 38
- LIST\_HEAD\_INITIALIZER macro, 38–39
- LIST\_REMOVE macro, 39
- loadable kernel module (LKM), 1
- lock
  - to ensure thread synchronization, 40–41
  - shared or exclusive, 40–41
- l\_read entry point, hooking, 35
- lstat system call, 30

## M

- make\_dev function, 16
- Makefile, 4–5
- malloc function, 73–74
- MALLOC macro, 74
- mbuf structure, 32
- memory allocation, 73–77
  - from user space, 77–81
- memory, detecting run-time patching, 125
- mi\_switch function, 123–124
- mkdir system call
  - debug message output from, 24
  - patching with inline function hook, 82–88
- modfind function, 10
- modid, for kernel module, 10
- modification time of file, preventing change, 107–114
- modstat function, 10
- module event handler, 2–3
- modules list, 103
- module\_stat structure, 11
- mtx\_lock function, 40
- mtx\_unlock function, 40
- mutexes, 40

## N

- name parameter
  - for DECLARE\_MODULE macro, 3
  - in SYSCALL\_MODULE, 8
- near call statement, 70
- <netinet/in\_pcb.h> header
  - struct in\_endpoints definition in, 52–53
  - u\_char inp\_vflag; definitions in, 53
- <netinet/tcp\_var.h> header, tcbinfo definition in, 53
- new\_sysent parameter, in SYSCALL\_MODULE, 9

## O

- objects, removing all references in kernel, 51
- offset parameter, in SYSCALL\_MODULE, 8
- offset value, for system call module, 8
- open system call, 30
- order parameter, for DECLARE\_MODULE macro, 3

## P

- padding, in argument structure, 7
- Perl, command-line execution, 12

- pfind function, 48
- Phrack* magazine, 90
- PIDHASH macro, 48
- pidhashtbl hash table, 74
- pid\_t p\_pid; field, in proc structure, 42
- port
  - finding hidden, 125
  - hiding open TCP-based, 52–56
- pr\_ctlinput entry point, in protocol switch table, 30–31
- pr\_ctloutput entry point, in protocol switch table, 30–31
- pread system call, 30
- preadv system call, 30
- pr\_init entry point, in protocol switch table, 30–31
- pr\_input entry point, in protocol switch table, 30–31
- printf, patching to invoke uprntf in place of, 72
- proc structure, 41–42
- processes
  - finding hidden, 123–124
  - hiding running, 41–46
    - example, 43–46
    - further steps, 46–51
- process\_hiding function, 48
- protocol switch table, 30
- protosw structure, 30–31
- pr\_output entry point, in protocol switch table, 30–31
- pwrite system call, 30
- pwritev system call, 30

## R

- read system call, 30
  - hooking, 26
- readv system call, 30
- rename system call, 30
- rmdir system call, 30
- rootkits
  - to bypass HIDSes, 91–117
    - execution redirection, 92–95
    - file hiding, 96–101
    - hiding KLD, 101–107
  - definition, xvi
  - detection, 119–126
    - design, 126
  - lack of need for unload routine, 95
  - new and improved example, 104–107
  - prevention, 126

- running kernel
  - loading and unloading code into, 5
  - userland code to patch, 63–90
- running processes, hiding, 41–46
  - further steps, 46–51

## S

- service system request, 6
- shared lock, 40–41
- signature, rootkit detection by, 119
- size parameter, for malloc function, 73
- size register\_t, for system call argument, 7
- stat system call, 30
- status of kernel module, function to return, 10
- struct cdev cdp\_c; structure, 60
- struct cdev\_priv, 60
- struct inpcbhead \*listhead field, in inpcbinfo structure, 53
- struct moduledata, definition, 3
- struct mtx inp\_mtx; field, in inpcb structure, 53
- struct mtx ipi\_mtx field, in inpcbinfo structure, 53
- struct mtx p\_mtx; field, in proc structure, 42
- struct vmSPACE \*p\_vmSPACE; field, in proc structure, 42
- struct\_in\_conninfo inp\_inc; field, in inpcb structure, 52
- sub parameter, for DECLARE\_MODULE macro, 3
- swapfile parameter, for kvm\_openfiles function, 64
- sx\_slock function, 40–41
- sx\_sunlock function, 41
- sx\_xlock function, 40–41
- sx\_xunlock function, 41
- symmetric multiprocessing (SMP), and data corruption, 39
- synchronization, of kernel queue data structures, 39–41
- <sys/conf.h> header
  - DEV\_MODULE macro definition in, 19
  - struct cdevsw definition in, 14
- /sys/fs/devfs/devfs\_devs.c file, 60
- /sys/i386/i386/trap.c file, 89
- /sys/kern/kern\_exec.c file, 92–95
- /sys/kern/kern\_exit.c file, 51
- /sys/kern/kern\_linker.c file, 102
- /sys/kern/kern\_module.c file, 103

- /sys/kern/vfs\_syscalls.c file, 96
- <sys/module.h> header
  - event handler function prototype in, 2
  - module\_stat structure definition in, 11
- <sys/mutex.h> header, Giant lock definition in, 102
- /sys/netinet/in\_proto.c file, 31–32
- <sys/proc.h> header, 41
  - PIDHASH macro definition, 48
  - pidhashtbl definition in, 47
  - proclist structure definition, 43
- <sys/protosw.h> header, protosw structure definition, 30–31
- <sys/queue.h> header, queue data definition, 38
- <sys/sysent.h> header
  - SYSCALL\_MODULE macro definition in, 8
  - sysent structure definition in, 7
  - system call function prototype in, 6
- syscall function, 11, 89
- SYSCALL\_MODULE macro, 8
- sysent structure, 7
- SYS\_mkdir constant, 25
- system call
  - common hooks, 29–30
  - hooking, 24–26
  - cloaking, 88–90
  - keystroke logging with, 26–28
- system call function, 6–7
- system call modules, 6–12
  - example, 9–10, 75–77
  - executing system call, 11
    - without C code, 12
  - finding hooks, 120–123
  - hiding running process,
    - example, 43–46
  - modfind function, 10
  - modstat function, 10
  - offset value, 8
  - overwriting, 77
  - syscall function, 11
  - SYSCALL\_MODULE macro, 8
  - sysent structure, 7
  - system call function, 6–7
- system call number, 8
- /sysufs/ufs/ufs\_vnops.c file, 108

## T

- TAILQ\_ENTRY(cdev\_priv) cdp\_list; field, 60
- TAILQ\_ENTRY(linker\_file) link; field, in struct linker\_file, 103

- tcbinfo.hashbase table, 125
- tcbinfo.ipi\_count variable, 125
- tcbinfo.listhead list, 53, 125
  - removing inpcb structure from, 54–56
- tcbinfo.parthashbase table, 125
- telnet, into remote machine, 56
- testing character device, 19–21
- time, changing, 108–111
- timestamps for files, preventing change, 107–114
- Transfer Control Protocol (TCP), hiding open port, 52–56
  - finding hidden, 125
- Tripwire, avoiding recognition by, 114–116
- truncate system call, 30
- type parameter, for malloc function, 73

## U

- u\_char inp\_vflag; field, in inpcb structure, 53
- ufs\_itimes function, 108–109
  - patching, 114
- UMA zone, checking for processes, 124
- unconditional jump, 81
- unlink system call, 30
- unload routine, rootkit lack of need for, 95
- uprntf, patching to invoke in place of printf, 72

- user space, 6*n*
  - allocating kernel memory from, 77–81
  - executing system call without program, 12
  - kernel space functions for data manipulation in, 12–13
  - program for executing system call, 11
- utime function, 108

## V

- Vaidheeswaran, Rajesh, 17
- virtual memory
  - in FreeBSD, 6*n*
  - interface for accessing, 63
  - patching code bytes, 66–70

## W

- Wehner, Stephanie, 93, 120
- write system call, 30
- writew system call, 30

## X

- x86 assembly, call statement, 70

## Z

- zombproc list, 42