

# Index

## A

### Access processing

- Rtl routine, 41
- Zw routine, 41

AddIndices, **function, 190–198**

AddNewKeyHandle, **function, 189–198**

AddRef, **function, 218–231**

AddTarget, **function, 260–262**

adjustData, **function, 78–96**

AdjustIndices, **function, 190–198**

AdjustNextNewIndex, **function, 190–198**

### ADS (Alternate Data Streams)

- file-hiding technique, 15–21
- syntax for, 20
- using, 277

AfterOriginalFunction, **function, 66–78**

Alert, **function, 260–262**

AllocateKeyHandle, **function, 190–198**

allocateUserMemory, **function, 66–78**

### alternate data streams (ADS)

- file-hiding technique, 15–21
- syntax for, 20
- using, 277

ANSI Prefix Manager (Pfx), **functional group, 40–41**

ANSI string table operations, **functional groups for hooking, 41**

anti-rootkit software, **types of, 254**

AntiHook, **anti-rootkit software, 254**

application programming, **injected function programming versus, 114**

ARP cache poisoning, **overview, 292**

autoloading, **overview, 138**

automatic updates, **rootkit prevention, 292**

autostarted application detection, **IceSword, 314**

## B

Background Intelligent Transfer Service (BITS), **using the, 5**

### Basic Rootkit

- configManager.c file code, 14–15
- configManager.h file code, 13
- fileManager.c file code, 17–19
- fileManager.h file code, 16
- Ghost.c file code, 10–12
- Ghost.h file code, 10
- SCMLoader.c file code, 22
- SCMUnloader.c file code, 25
- summary, 26

beforeEncode

- function, 66–78
- process injection hook, 67–78

BeforeOriginalFunction, **function, 66–78**

BITS (Background Intelligent Transfer Service), **using the, 5**

### blocking

- PGP encoding, 99–100
- unexpected operations, 298

blue screen of death (BSOD), **defined, 169**

browser helper object detection, **IceSword, 314**

BSOD (blue screen of death), **defined, 169**

build command, **SCMUnloader.c file, 25**

## build environment, problems, 23

buildController.bat file, I/O Processing, 107

## C

**C compiler, downloading a, 1**

**C# Visual Studio, installing, 120**

CALL\_DATA\_STRUCT, members of, 63

callType, CALL\_DATA\_STRUCT, 63

CClientExtension, function, 219–231

checkConnectionButton\_Click, function, 262–268

**Checked DDK shell, using the, 5**

checkPattern, function, 54–63

**cleanup, installation, 251–254**

**client operations, functional groups for hooking, 39**

**Client Server Run Time (Csr), functional group, 39**

CloseTDIConnection, function, 122–130

CMessageEvents, function, 218–231

### code

commManager.c file, 122–133

commManager.h file, 121–122

configManager.c file, 14–15

configManager.h file, 13

ControlForm.cs file, 263–268

Controller.c file, 105–106

in data segment prevention technique, 299

directory hiding, 203–205

fileManager.c file, 17–19

fileManager.h file, 16

filterManager.c file, 142–145, 173–174

filterManager.h file, 142

Ghost.c file, 10–12, 33–36, 146–150, 172–173, 198

Ghost.h file, 10, 51

GhostTracker.cs file, 260–262

HideMe.c file, 206–210, 211

Hook Function, 31–33

hookManager.c file, 36–37, 55–63, 199–202

hookManager.h file, 37–38, 52–54, 198–199

injectManager.c file, 67–78

injectManager.h file, 63–66

installation, 246–247, 249–251, 251–254

IoManager.c file, 110–114, 154–166, 174

IoManager.h file, 106–107, 150–154

Kernel Memory Protection, 28–30

keyManager.c file, 176–184

keyManager.h file, 174–175

link library, 44–46

Listen.cs file, 271–272

Lotus Notes Client Extension testing, 242

LotusExtension.c file, 235–239

LotusExtension.def file, 240

LotusExtension.h file, 234

LotusExtension.mak file, 240

Mozilla Firefox installation, 249–251

Outlook Client Extension testing, 232

OutlookExtension.cpp file, 219–230

OutlookExtension.h file, 216–218

parse86.c file, 79–96

parse.c file, 79–96

parse.h file, 78

peFormat.h file, 97–99

readme.txt file, 241

registryManager.c file, 190–197

registryManager.h file, 188–189

SCMLoader.c file, 22

SCMUnloader.c file, 25

TargetController.cs file, 269–270

### code (Basic Rootkit)

configManager.c file, 14–15

configManager.h file, 13

fileManager.c file, 17–19

fileManager.h file, 16

Ghost.c file, 10–12

SCMLoader.c file, 22

SCMUnloader.c file, 25

### code (Communications)

commManager.c file, 122–133

commManager.h file, 121–122

### code (Concealment)

Ghost.c file, 198

HideMe.c file, 206–210, 211

hookManager.c file, 199–202

hookManager.h file, 198–199

registryManager.c file, 190–197

registryManager.h file, 188–189

### code (E-mail Filtering)

LotusExtension.c file, 235–239

LotusExtension.def file, 240

LotusExtension.h file, 234

LotusExtension.mak file, 240

OutlookExtension.cpp file, 219–230

OutlookExtension.h file, 216–218

readme.txt file, 241

**code (Filter Drivers)**

filterManager.c file, 142–145  
 filterManager.h file, 142  
 Ghost.c file, 146–150  
 IoManager.c file, 154–166  
 IoManager.h file, 150–154

**code (Ghost Tracker)**

ControlForm.cs file, 263–268  
 GhostTracker.cs file, 260–262  
 Listen.cs file, 271–272  
 TargetController.cs file, 269–270

**code (I/O Processing)**

Controller.cs file, 105–106  
 IoManager.c file, 110–114  
 IoManager.h file, 106–107

**code (Kernel Hooks)**

filterManager.c file, 173–174  
 Ghost.c file, 33–36  
 hookManager.c file, 36–37  
 hookManager.h file, 37–38  
 IoManager.c file, 174  
 keyManager.c file, 176–184  
 keyManager.h file, 174–175

**code (Key Logging), Ghost.c file, 172–173****code (User Hooks)**

Ghost.c file, 51–52  
 Ghost.h file, 51  
 hookManager.c file, 55–63  
 hookManager.h file, 52–54  
 injectManager.c file, 67–78  
 injectManager.h file, 63–66  
 parse86.c file, 79–96  
 parse86.h file, 78  
 peFormat.h file, 97–99

**combined filtering, diagrammed, 141****comint32.sys, rootkit/device driver, 21****comint32, debug statements and, 13****command**

build, 25  
 ipconfig, 121

**Command Prompt window, VCVARS32.BAT, 23****commManager.c file**

code, 122–133  
 functions list, 122  
 used in Communications, 122–133

**commManager.h file**

code, 121–122  
 used in Communications, 121–122

**Communications**

code, 121–133  
 commManager.c file, 122–130  
 commManager.h file, 121–122  
 example, 120–133  
 initiating the connection, 120  
 running the example, 133–135  
 SOURCES, 130–131  
 summary, 135–136  
 Transport Driver Interface (TDI), 119–120

**compiling, programs, 21, 23–24****completion routine, keyboard I/O, 168****Compression and decompression operations, Rtl routine, 41****computer code. See code****Concealment**

directory hiding, 203–205  
 directory hiding code, 203–205  
 Ghost.c file, 198  
 Ghost.c file code, 198  
 HideMe.c file code, 206–210, 211  
 hookManager.c file, 199–202  
 hookManager.c file code, 199–202  
 hookManager.h file, 198–199  
 hookManager.h file code, 198–199  
 overview, 187  
 process hiding, 205–211  
 registry key hiding, 187–202  
 registryManager.c file, 189–198  
 registryManager.c file code, 190–197  
 registryManager.h file, 188–189  
 registryManager.h file code, 188–189  
 summary, 212–213  
 testing, 211–212

**configManager.c file, code, 14–15****configManager.h file**

code, 13  
 DriverEntry function, 13

**configuration file**

creating the, 23  
 diagrammed, 16

**connection**

initiating the, 120  
 rootkit controller, 257

**Console Application, using the, 105–114****Control categories, overview, 257****Control Panels, control category, 257**

# ControlForm

---

## ControlForm

- function, 262–268
- overview, 273
- rootkit remote controller implementation, 273

## ControlForm.cs file

- code, 263–268
- functions list, 262
- rootkit remote controller implementation, 262–268

## controller

- Control categories, 257
- designing the, 256–257
- determining the necessity of a, 255
- interface, 256
- Interface medium, 256
- Summary view, 257

## Controller.c file

- code, 105–106
- I/O Processing, 105–106

## CreateFileW, function, 50–51

## CreateHiddenKeyIndices, function, 190–198

## createTrampoline, function, 66–78

## creating

- a basic rootkit, 9–12
- configuration files, 23

## CrsNewThread, routine, 39

## Csr (Client Server Run Time), functional group, 39

## CsrCaptureMessageBuffer, routine, 39

## CsrClientCallServer, routine, 39

## CsrConnectClientToServer, routine, 39

# D

## Dbg (Debug Manager), functional group, 39

## DbgBreakPoint, routine, 39

## DbgPrint, routine, 39

## DbgPrint statements, Ghost.c file and, 13

## DbgUiConnectToDbg, routine, 39

## DbgUserBreakPoint, routine, 39

## Debug Manager (Dbg), functional group, 39

## debug operations, functional groups for hooking, 39

## debug statements

- comint32 and, 13
- Ghost.c file and, 13

## Debug View

- downloading, 5
- output, 24
- utility, 2

## Debugging Tools for Windows

- downloading, 2
- verifying, 7

## DebugView, freeware, 301–302

## DeleteMessage, function, 219–231

## demand start loading, defined, 21

## DeregisterEntry, function, 234–239

## detecting, rootkits, 275–290

## detection methods

- IceSword, 312–313
- rootkit, 275–279

## detection software. *See also* software

- F-Secure Blacklight, 281–282
- IceSword, 283–286
- Rootkit Hook Analyzer, 282–283
- RootkitRevealer, 280–281
- Strider GhostBuster (monofont ghostbuster), 280
- Sophos Anti-Rootkit, 286–287

## DetourFunction, function, 66–78

## device driver

- comint32.sys, 21
- diagrammed, 12
- handling IO within the, 107–114
- loading a, 21–22
- rootkit, 9–15
- unloading a, 21, 25

## device extension, defined, 138

## device pointer

- newFileSysDevice, 146–150
- newNetworkDevice, 146–150
- oldFileSysDevice, 146–150
- oldNetworkDevice, 146–150

## DeviceIoControl, function, 103–104

## diagrams

- basic IO control, 104
- combined filtering, 141
- configuration file, 16
- device driver, 12
- file system filters, 139
- GhostTracker threading model, 259
- key logger insertion, 169
- key logger synchronization, 170
- key processing, 171
- loading/unloading a device driver, 21
- Memory Descriptor Lists, 28
- network filtering, 140

NewSystemCallTable, 30  
 parsing x86 instructions, 96  
 PGP Monitor for Windows 2000, XP and, 2003, 101  
 process hiding, 206  
 Process Hiding Detection, 279  
 rootkit environment, 134  
 SwapContext Process Hiding Detection, 279  
 system call table, 30  
 system call table hooking, 31  
 trampoline process, 49  
 ZwMapViewOfSection, 44

**dialog box**

Filter, 303  
 IDA file selection, 7  
 Load File, 307  
 Recipient Selection, 115–116  
 Save As, 115–116  
 Save PGP Zip As, 115–116  
 Windows Firewall, 293

**directory hiding**

coding, 203–205  
 overview, 203–205

**Diskmon, utility, 2, 5–6**

Dispose, **function, 260–268**  
 DllMain, **function, 218–231, 234–239**

**downloading**

C compiler, 1  
 Debug View, 5  
 Debugging Tools for Windows, 2  
 Lotus Notes C API, 233  
 Microsoft Driver Development Kit (DDK), 2  
 Microsoft Visual C++ 2005 Express, 2  
 PGP Professional Version, 9, 99  
 symbols, 2–3  
 Windows Platform Software Development Kit (SDK), 1–2

**driver load prevention, prevention technique, 298–299****DRIVER\_DATA, operating system structure, 10–12**

DriverEntry  
 entry function, 10–12, 13  
 function, 210  
 I/O Processing, 110  
 drivers.exe, **rootkit testing with, 26**  
 DriverUnload, **function, 34**  
 Dynamic Link Libraries (DLLs), **overview, 43–44**

**E****E-mail Filtering**

code for testing the Lotus Notes Client  
 Extension, 242  
 code for testing the Outlook Client Extension, 232  
 EXCHEXT.H skeletal file, 216  
 installing a Lotus Notes client filter, 241–242  
 installing an Outlook client filter, 231  
 Lotus files, 233  
 Lotus Notes, 232–241  
 LotusExtension.c file code, 235–239  
 LotusExtension.c implementation file,  
 232, 234–239  
 LotusExtension.def file code, 240  
 LotusExtension.def implementation file,  
 233, 240  
 LotusExtension.h file code, 234  
 LotusExtension.h implementation file,  
 232, 234  
 LotusExtension.mak file code, 240  
 LotusExtension.mak implementation file,  
 233, 240  
 Microsoft Outlook, 215–231  
 OutlookExtension.cpp file code, 219–230  
 OutlookExtension.cpp implementation file,  
 216, 218–231  
 OutlookExtension.dsp skeletal file, 216  
 OutlookExtension.dsw skeletal file, 216  
 OutlookExtension.h file code, 216–218  
 OutlookExtension.h implementation file,  
 216, 218  
 readme.txt file code, 241  
 Readme.txt implementation file, 233  
 Readme.txt skeletal file, 216  
 Stdafx.cpp skeletal file, 216  
 Stdafx.h skeletal file, 216  
 summary, 242  
 testing the Lotus Notes client extension, 242  
 testing the Outlook client extension, 231–232  
**End User License Agreements (EULAs), overview, 244–245**  
 EndOfInjectedCode, **function, 66–78**  
**environment variables, modifying, 23**  
**Etw (Event Tracing for Windows), functional group, 40**  
**EtwEnableTrace, routine, 40**

**EtwGetTraceEnableFlags, routine, 40**

**EtwGetTraceEnableLevel, routine, 40**

**EtwTraceEvent, routine, 40**

**EULAs (End User License Agreements), overview, 244–245**

**Event History, control category, 257**

**Event operations, Zw routine, 41**

**Event Status, control category, 257**

**Event Tracing for Windows (Etw), functional group, 41**

**example**

Communication, 120–133

File Filtering, 141–166

Kernel Hooks, 33–38

Key Logging, 171–185

Rootkit Controller, 258–273

testing the Key Logging, 185

**ExchEntryPoint, function, 216, 218–231**

**EXCHEXT.H, E-mail filtering skeletal file, 216**

**ExInterlockedInsertTailList, function, 170**

**ExInterlockedRemoveHeadList, function, 170**

## F

**F-Secure Blacklight**

anti-rootkit software, 254

detection software, 281–282

freeware, 311

**fail-open functionality, fail-safe functionality versus, 244**

**feedback, types of, 244**

**file**

configuration, 23

filtering, 138–139

functions differentiated, 20

parsing a PE formatted, 97–99

tagging a tracked, 277

**file-hiding**

alternate data streams technique, 15–21

technique, 15–21

testing, 212

**File operations, Zw routine, 41**

**File and Registry (Zw), functional group, 41**

**file system filtering**

diagrammed, 139

performing, 138

**file system tamper detection, IceSword, 314**

**fileManager.c file**

code, 17–19

GetFile (mf) function, 17–19

PutFile function, 17–19

**fileManager.h file**

code, 16

functions used in, 16

MASTER\_FILE (mf), 16

**FileMon. See also FileMonitor**

utility, 2, 5–6

**FileMonitor**

freeware, 304–305

RegistryMonitor versus, 305

**FileName, Unicode string, 20**

**filter, adding a keyboard, 168–170**

**Filter dialog box, 303**

**Filter Drivers**

combined filtering, 140–141

defined, 137

example, 141–166

file filtering, 138–139

filtermanager.c file code, 142–145

filterManager.h file code, 142

Ghost.c file code, 146–150

inserting a, 137–138

IoManager.c file code, 154–165

IoManager.h file code, 150–154

network filtering, 139–140

SOURCES, 166

summary, 166

**filtering**

combined, 140–141

file, 138–139

network, 139–140

**filtering software. See intended installation**

**filterManager.c file**

code, 142–145, 173–174

filter drivers, 142–145

functions list, 142

key logging, 173–174

**filterManager.h file**

code, 142

filter drivers, 142

key logging, 174

**FindKeyHandle, function, 189–198**

**findProcess, function, 210**

**findUnresolved, function, 54–63**

**forensic data, feedback, 244****Forensics, control category, 257**FreeKernelAddress, **function, 54–63**FreeKeyHandle, **function, 190–198**FreeKeyTrackingData, **function, 189–198****freeware**

DebugView, 301–302  
 F-Secure Blacklight, 311  
 FileMonitor, 304–305  
 IceSword, 312–314  
 IDA, 306–307  
 RegistryMonitor, 302–304  
 Rootkit Hook Analyzer, 311–312  
 Rootkit Unhooker, 308–310  
 RootkitRevealer, 310  
 Samurai, 307–308  
 Sophos Anti-Rootkit, 315  
 TCPView, 305

**function**

defining a hook, 31–33  
 trampoline, 48–49

**function (Basic Rootkit)**

DriverEntry, 10–12, 13  
 GetFile, 16, 17–19, 20  
 PutFile, 16–19, 20

**function (Communications)**

CloseTDIConnection, 122–130  
 OpenTDIConnection, 122–130  
 SendToRemoteController, 122–130  
 TDICompletionRoutine, 122–130  
 TimerDPC, 122–130

**function (Concealment)**

AddIndices, 190–198  
 AddNewKeyHandle, 189–198  
 AdjustIndices, 190–198  
 AdjustNextNewIndex, 190–198  
 AllocateKeyHandle, 190–198  
 CreateHiddenKeyIndices, 190–198  
 DriverEntry, 210  
 FindKeyHandle, 189–198  
 findProcess, 210  
 FreeKeyHandle, 190–198  
 FreeKeyTrackingData, 189–198  
 GetKeyName, 202  
 GetNewIndex, 190–198  
 GetPointerByHandle, 202  
 GetSubkeyCount, 190–198

InitializeKeyTracking, 189–198

NewZwEnumerateKey, 202

NewZwOpenKey, 202

NewZwQueryKey, 202

OnDeviceControl, 210

**function (E-mail Filtering)**

AddRef, 218–231, 219–231  
 CClientExtension, 219–231  
 CMessageEvents, 218–231  
 DeleteMessage, 219–231  
 DeregisterEntry, 234–239  
 DllMain, 218–231, 234–239  
 ExchEntryPoint, 216, 218–231  
 Install, 219–231  
 LogAttachments, 219–231  
 LogBody, 219–231  
 LogContent, 219–231, 234–239  
 MainEntryPoint, 232, 234–239  
 OnCheckNames, 218–231  
 OnCheckNamesComplete, 219–231  
 OnRead, 218–231  
 OnReadComplete, 218–231  
 OnSendMail, 234–239  
 OnSubmit, 216, 219–231  
 OnSubmitComplete, 216, 219–231  
 OnWrite, 218–231  
 OnWriteComplete, 216, 218–231  
 ParseRecipientList, 234–239  
 QueryInterface, 218–231  
 RegisterEntry, 234–239  
 Release, 218–231  
 SaveAttachments, 234–239  
 SaveBody, 234–239  
 SaveRecipients, 234–239

**function (Filter Drivers)**

insertFileFilter, 142–145  
 insertNetworkFilter, 142–145  
 IoAttachDeviceToDeviceStack, 138  
 IoAttachDeviceToDeviceStackSafe, 138  
 removeFilter, 142–145

**function (Ghost Tracker)**

AddTarget, 260–262  
 Alert, 260–262  
 checkConnectionButton\_Click, 262–268  
 ControlForm, 262–268  
 Dispose, 260–268  
 InitializeComponent, 262–268

## function (Ghost Tracker) (continued)

---

### function (Ghost Tracker) (continued)

- Listen, 270–272
- Main, 260–262
- MainForm, 260–262
- Ping, 269–270
- Start, 268–272
- Stop, 269–272
- TargetController, 268–270
- targetListView\_SelectedIndexChanged, 260–262

### function (I/O Processing), DeviceIoControl, 103–104

#### function (Kernel Hooks)

- DriverUnload, 34
- Hook, 36–37
- InterlockedExchange, 30

#### function (Key Logging)

- ExInterlockedInsertTailList, 170
- ExInterlockedRemoveHeadList, 170
- GetKey, 184
- InitializeListHead, 170
- InitializeLogThread, 184
- insertKeyboardFilter, , 173–174
- KeInitializeSemaphore, 170
- KeInitializeSpinLock, 170
- KeWaitForSingleObject, 170
- KeyLoggerThread, 185
- OnCancel, 185
- OnKeyboardRead, 184
- OnReadCompletion, 184
- OnUnload, 172–173
- PsCreateSystemThread, 170
- PsTerminateSystemThread, 170
- StartKeylogger, 174, 185
- StopKeylogger, 185

#### function (User Hooks)

- adjustData, 78–96
- AfterOriginalFunction, 66–78
- allocateUserMemory, 66–78
- beforeEncode, 66–78
- BeforeOriginalFunction, 66–78
- checkPattern, 54–63
- CreateFileW, 50–51
- createTrampoline, 66–78
- DetourFunction, 66–78
- EndOfInjectedCode, 66–78
- findUnresolved, 54–63
- FreeKernelAddress, 54–63

- GetFunctionAddress, 54–63
- getHookPointers, 66–78
- GetImageSize, 54–63
- getNextInstruction, 78, 78–96
- getx86Instruction, 66–78
- hookFunction, 63
- HookKernel, 54–63
- HookTable, 66–78
- isJump, 78–96
- IsSameFile, 44–47, 54–63
- IsSameString, 54–63
- lstrcmpiW, 50–51
- makeWritable, 66–78
- MapKernelAddress, 54–63
- NewZwMapViewOfSection, 54–63
- noTransferOp, 78–96
- processInject, 66–78
- transferData, 78–96
- transferDataPrefix, 78–96
- transferInstruction, 78–96
- transferOp0F, 78–96
- transferOp66, 78–96
- transferOp67, 78–96
- transferOpF6, 78–96
- transferOpF7, 78–96
- transferOpFF, 78–96

#### functional groups

- ANSI Prefix Manager (Pfx), 40–41
- client operations, 39
- Client Server Run Time (Csr), 39
- Debug Manager (Dbg), 39
- Event Tracing for Windows (Etw), 41
- File and Registry (Zw), 41
- Kernel (Ki), 40
- Loader Manager (Ldr), 40
- server operations, 39

#### functions

- of GetFile, 20
- in hookManager.c file, 54–55
- in injectManager.c file, 66–78
- mapping, 20
- in ntdll.dll, 39
- of parse86.c file, 78–96
- of parse86.h file, 78
- of PutFile, 20
- resource, 20
- of Rootkit Unhooker, 308
- types of, 20

**G**

GetFile, **function, 16, 17–19, 20**  
 GetFunctionAddress, **function, 54–63**  
 getHookPointers, **function, 66–78**  
 GetImageSize, **function, 54–63**  
 GetKey, **function, 184**  
 GetKeyName, **function, 202**  
 GetNewIndex, **function, 190–198**  
 getNextInstruction, **function, 78, 78–96**  
 GetPointerByHandle, **function, 202**  
 GetSubkeyCount, **function, 190–198**  
 getx86Instruction, **function, 66–78**  
**Ghost**  
   rootkit example, 9–15  
   using to block PGP encoding, 99–100  
**Ghost Tracker**  
   ControlForm.cs file code, 263–268  
   GhostTracker.cs file code, 260–262  
   Listen.cs file code, 271–272  
   TargetController.cs file code, 269–270  
**Ghost.c file**  
   code for Basic Rootkit, 10–12  
   code for Concealment, 198  
   code for Filter Drivers, 146–150  
   code for Kernel Hooks, 33–36  
   code for Key Logging, 172–173  
   code for User Hooks, 51  
   comint32, 13  
   concealment, 198  
   DbgPrint statements, 13  
   debug statements, 13  
   device pointers, 146  
   DriverEntry function, 10–12  
   DriverUnload function, 34  
   filter drivers, 146–150  
   kernel32Base variable, 51–52  
   key logging, 172–173  
   NewSystemCallTable variable, 33–36  
   OldZwMapViewOfSection variable, 33–36  
   OnUnload function, 10  
   pMyMDL variable, 33–36  
   ZwProtectVirtualMemory, 51–52  
   ZwProtectVirtualMemory variable, 51–52  
**Ghost.h file**  
   Basic Rootkit code, 10  
   CreateFileW function, 50–51  
   DRIVER\_DATA, 10–12

  lstrcmplW function, 50–51  
   OnUnload function, 10–12  
   user hooks, 50–51  
   User Hooks code, 51  
**GhostTracker, controller, 120–121**  
**GhostTracker form**  
   overview, 273  
   rootkit remote controller implementation, 273  
**GhostTracker threading model, diagram, 259**  
**GhostTracker.cs file**  
   code, 260–262  
   functions list, 260  
   rootkit remote controller implementation, 260–262  
**global variable, listOffset, 210–211**

**H**

**hardening**  
   defined, 295  
   Samurai HIPS, 295–297  
**Heap operations, Rtl routine, 41**  
**HideMe.c**  
   code, 206–210, 211  
   file, 206–211  
**HOOK, macro, 37–38**  
**Hook Function**  
   code, 31–33  
   defining a, 31–33, 47–48  
**Hook function, hookManager.c file, 36–37**  
**hookFunction, CALL\_DATA\_STRUCT, 63**  
**HOOK\_INDEX, macro, 37–38**  
**hooking, problems with, 42**  
**HookKernel, function, 54–63**  
**hookManager.c file**  
   checkPattern function, 55–63  
   code, 36–37, 55–63, 199–202  
   concealment, 199–202  
   findUnresolved function, 55–63  
   FreeKernelAddress function, 54–63  
   functions in, 54–55  
   functions list, 54–55  
   GetFunctionAddress function, 55–63  
   GetImageSize, function, 55–63  
   Hook function, 36–37  
   HookKernel function, 54–63  
   IsSameFile function, 54–63  
   IsSameString function, 54–63  
   kernel hooks, 36–37

### hookManager.c file (continued)

MapKernelAddress function, 54–63  
NewZwMapViewOfSection function, 36–37, 54–63  
user hooks, 54–63

### hookManager.h file

code, 37–38, 52–54, 198–199  
concealment, 198–199  
global variables, 37  
kernel hooks, 37–38  
KeServiceDescriptorTable, 37  
NewZwMapViewOfSection, 37–38  
ServiceDescriptorEntry, 37–38  
user hooks, 52–54

### HookTable, function, 66–78

### Host-based Intrusion Prevention Systems

blocking unexpected operations, 298  
hardening, 295–297  
virtualizing, 297

### I/O control, testing, 114–117

### I/O functions, differentiated, 20

### I/O Processing

buildController.bat, 107  
code for handling IO within the device driver,  
108–110  
Controller.c file, 105–106  
Controller.c file code, 105–106  
DeviceIoControl function, 103–104  
DriverEntry, 110  
handling IO within the Device Driver, 107–114  
injected function programming, 114  
IoManager.c file, 110–112  
IoManager.c file code, 110–114  
IoManager.h file, 106–107  
IoManager.h file code, 106–107  
SOURCES, 112  
summary, 117–118  
testing I/O control, 114–117  
The Console Application, 105–107

### I/O Request Packets (IRPs), defined, 120

### IceSword

autostarted application detection, 314  
browser helper object detection, 314  
detection methods, 312–313  
detection software, 283–286  
file system tamper detection, 314

freeware, 312–314  
kernel module detection, 313  
kernel system call table hook detection, 314  
message hook detection, 314  
process creation detection, 314  
process termination detection, 314  
registry tamper detection, 314  
service detection, 314  
winsock catalog entry detection, 314

### IDA

advantages to using, 47–48  
downloading, 2  
file selection dialog box, 7  
freeware, 306–307  
overview, 6  
plug-ins availability, 306  
process injection and, 47–48  
verifying, 7

index, CALL\_DATA\_STRUCT, 63

### initialization files, installation, 248–249

InitializeComponent, function, 262–268  
InitializeKeyTracking, function, 189–198  
InitializeListHead, function, 170  
InitializeLogThread, function, 184

### Initializing and using critical selections, Rtl routine, 41

### Initializing and using resources, Rtl routine, 41

### Initializing and using security objects, Rtl routine, 41

### Initializing and using strings, Rtl routine, 41

### Initializing and using threads, Rtl routine, 41

### injected function programming

application programming versus, 114  
overview, 114

### injectManager.c file

code, 67–78  
functions listed, 66–67  
functions of, 66–67  
process injection and, 66–78  
user hooks, 66–78

### injectManager.h file

CALL\_DATA\_STRUCT members, 63  
code, 63–66  
Ghost.h file, 50–51  
user hooks, 63–66

### IN\_PROCESS\_DATA, Ghost.h file, 50–51

### insertFileFilter, function, 142–145, 145

### inserting, filter drivers, 137–138

### insertKeyboardFilter, function, 173–174

`insertNetworkFilter`, **function**, 142–145

`Install`, **function**, 219–231

#### installation

- cleanup, 251–254
- code, 246–247, 249–254
- considerations summary, 254
- End User License Agreements (EULAs), 244–245
- initialization files, 248–249
- intended, 243–244
- persistence, 245–246
- privilege escalation, 245
- registry settings, 247–248
- software for intended, 244
- technique for Mozilla Firefox (mf), 249–251
- testing, 254
- through exploitation, 249–251
- unintended, 245
- using `ZwSetSystemInformation`, 246–247

#### installing

- C# Visual Studio, 120
- a Lotus Notes client filter, 241–242
- the Microsoft Driver Development Kit (DDK), 4
- Microsoft Visual C++ 2005 Express, 5
- an Outlook client filter, 231
- a rootkit, 21–25
- the Windows Platform Software Development Kit (SDK), 5

`InstallShield`, **software**, 244, 287

**instruction, parsing x86**, 96

**integrating, the SQL Server**, 5

#### intended installation

- overview, 243–244
- software, 244

**intercept method, IRP**, 169–170

**interface, overview**, 256

**interface-driven, low-level technology versus**, 256

**Interface medium, overview**, 256

`InterlockedExchange`, **function**, 30

## IO

- control basic overview, 104
- handling within the device driver, 107–114

`IoAttachDeviceToDeviceStack` **function**,  
`IoAttachDeviceToDeviceStackSafe`  
**function versus**, 138

#### `IoManager.c` file

- code for Filter Drivers, 154–166
- code for I/O Processing, 106–107, 110–114

- code for Key Logging, 174

- filter drivers, 154–165

- I/O Processing, 110–112

- key logging, 174

#### `IoManager.h` file

- code, 150–154

- filter drivers, 150–154

- I/O Processing, 106–107

**IP address, finding an**, 121

`ipconfig` **command, finding an IP address with the**, 121

**IPv4 and IPv6 operations, Rtl routine**, 41

**IRP intercept method, key logging and the**, 169–170

**IRPs (I/O Request Packets), defined**, 120

`IRQL = APC_LEVEL`, **processing level**, 168

`IRQL = DIRQL`, **processing level**, 168

`IRQL = DISPATCH_LEVEL`, **processing level**, 168

`IRQL = PASSIVE_LEVEL`, **processing level**, 168

`isJump`, **function**, 78, 78–96

**ISO image, downloading the**, 1–2

`IsSameFile`, **function**, 44–47, 54–63

`IsSameString`, **function**, 54–63

## K

`KeInitializeSemaphore`, **function**, 170

`KeInitializeSpinLock`, **function**, 170

**Kerio Personal Firewall, overview**, 294

#### kernel

- call table hooking detection, 277
- function hooking detection, 277
- hooking problems, 42

**Kernel Debugger, overview**, 6

**kernel hook prevention, prevention technique**, 298

#### Kernel Hooks

- basic components of, 31
- code for defining a hook function, 31–33
- code for kernel memory protection, 29–30
- `DriverUnload` function, 34
- example, 33–38
- functional groups, 39–41
- `Ghost.c` file code, 33–36
- `hookManager.c` file, 36–37
- `hookManager.c` file code, 36–37
- `hookManager.h` file, 37–38
- `hookManager.h` file code, 37–38
- kernel hook functions, 31–33
- kernel hook macros, 30–31

### Kernel Hooks (continued)

- kernel memory protection, 28–31
- problems with, 42
- summary, 42
- system service table, 27–28

### Kernel (Ki), functional group, 40

#### kernel memory, scanning, 278

#### kernel mode device driver, wOpenFile, 20

#### kernel module detection, IceSword, 313

#### kernel system call table hook detection, IceSword, 314

kernel32Base **variable**, Ghost.c, 51–52

kernel32.dll, Ghost.h, 50–51

KeServiceDescriptorTable

- hookManager.h file, 37–38

- system call table, 27–30

KeWaitForSingleObject, **function**, 170

#### key code mapping, key processing versus, 171

#### key codes, interpreting, 170–171

#### key logger

- insertion diagram, 169

- synchronization diagram, 170

#### Key Logging

- example, 171–185

- example testing, 185

- filterManager.c file, 173–174

- filterManager.c file code, 173–174

- filterManager.h file, 174

- Ghost.c file, 172–173

- Ghost.c file code, 172–173

- IoManager.c file, 174

- IoManager.c file code, 174

- IRP intercept method, 169–170

- key codes, 170–171

- keyboard filter, 168–170

- keyManager.c file, 176–184

- keyManager.c file code, 176–184

- keyManager.h file, 174–175

- keyManager.h file code, 174–175

- processing levels, 167–168

- SOURCES, 172

- summary, 186

- threading and synchronization, 170

#### key processing

- diagram, 171

- key code mapping versus, 171

#### keyboard filter, adding a, 168–170

#### keyboard I/O, completion routine, 168

keyboardData **global variable**, key logging, 172–173

KeyLoggerThread, **function**, 185

keyManager.c **file**

- code, 176–184

- key logging, 176–184

keyManager.h **file**

- code, 174–175

- key logging, 174–175

Ki (Kernel), **functional group**, 40

KiRaiseUserExceptionDispatcher, **routine**, 40

KiUserApcDispatcher, **routine**, 40

KiUserCallbackDispatcher, **routine**, 40

KiUserExceptionDispatcher, **routine**, 40

known good environment, **defined**, 276

## L

Ldr (Loader manager), **functional group**, 40

LdrGetDllHandle, **routine**, 40

LdrGetProcedureAddress, **routine**, 40

LdrInitializeThunk, **routine**, 40

LdrLockLoaderLock, **routine**, 40

LdrUnlockLoaderLock, **routine**, 40

library, finding a specific, 44–47

link heartbeats, **feedback**, 244

link library, **code**, 44–46

Listen, **function**, 270–272

Listen.cs **file**

- code, 271–272

- functions list, 270

- rootkit remote controller implementation, 270–272

listOffset, **global variable**, 210–211

Load File dialog box, 307

Loader Manager (Ldr), **functional group**, 40

loader operations, **functional groups for hooking**, 40

#### loading

- demand start, 21

- the rootkit, 24

Local Kernel Debugger, **opening the**, 7

LogAttachments, **function**, 219–231

LogBody, **function**, 219–231

LogContent, **function**, 219–231, 234–239

Lotus files, **E-mail filtering**, 233

#### Lotus Notes

- Client Extension testing code, 242

- E-mail filtering overview, 232–233

- installing a Lotus Notes client filter, 241–242

LotusExtension.c file, 234–239  
 LotusExtension.def file, 240  
 LotusExtension.h file, 234  
 LotusExtension.mak file, 240  
 readme.txt file, 241  
 testing the Lotus Notes client extension, 242

### Lotus Notes C API, downloading, 233

LotusExtension.c  
 code, 235–239  
 E-mail filtering, 234–239  
 E-mail filtering implementation file, 232  
 functions list, 234  
 LotusExtension.def  
 code, 240  
 E-mail filtering, 240  
 E-mail filtering implementation file, 232  
 LotusExtension.h  
 code, 234  
 E-mail filtering implementation file, 232, 234  
 LotusExtension.mak  
 code, 240  
 E-mail filtering, 240  
 E-mail filtering implementation file, 232

### low-level technology, interface-driven versus, 256

lstrcmpiW, function, 50–51

## M

### macros, hooking, 30–31, 37–38

Main, function, 260–262  
 MainEntryPoint, function, 232, 234–239  
 MainForm, function, 260–262  
 MAKEFILE file, content of, 20  
 makeWritable, function, 66–78  
 Manipulating data types, Rtl routine, 41  
 Manipulating memory, Rtl routine, 41  
 MapKernelAddress, function, 54–63  
 mapping functions, differentiated, 20  
 MASTER\_FILE, ADS location, 16  
 MDLFlags, Memory Descriptor List (MDL) and, 29–30  
 Memory Descriptor List (MDL)  
 defined, 28  
 diagrammed, 28  
 MDLFlags and, 29–30  
 ntddk.h, 28–29  
 using, 28–30

### memory scanning, overview, 278

### message hook detection, IceSword, 314

### MetaSploit software, using, 8

### Microsoft, website, 1–2

### Microsoft Driver Development Kit (DDK)

downloading the, 1–2  
 installing the, 4  
 shortcuts, 4–5  
 verifying the, 6

### Microsoft MSDN subscription, necessity of having a, 1–2

### Microsoft Outlook

E-mail filtering overview, 215–216  
 installing an Outlook client filter, 231  
 OutlookExtension.cpp file, 218–231  
 OutlookExtension.h file, 216–218  
 testing the Outlook client extension, 231–232

### Microsoft Visual C++ 2005 Express

downloading, 2  
 installing, 5  
 verifying, 6

### Microsoft Windows 2000, XP, and 2003, PGP Monitor, 101

### modifying, environment variables, 23

### Monitor History, control category, 257

### Monitor Status, control category, 257

### Mozilla Firefox, installation technique for, 249–251

### MSDN, integrating, 5

## N

### network, filtering, 139–140

### network filters, diagrammed, 140

### newFileSysDevice, device pointer, 146–150

### newKeyboardDevice global variable, key logging, 172–173

### newNetworkDevice, device pointer, 146–150

### NewSystemCallTable

Ghost.c file variable, 33–36  
 hookManager.h file variable, 37–38  
 system call table diagrammed, 30

### NewZwEnumerateKey, function, 202

### NewZwMapViewOfSection function

function, 54–63  
 hookManager.c file, 36–37  
 hookManager.h file, 37–38  
 process injection, 47

### NewZwOpenKey, function, 202

# NewZwQueryKey, function

---

NewZwQueryKey, **function**, 202  
noTransferOp, **function**, 78–96  
ntddk.h  
    defining a hook function, 47  
    Memory Descriptor List, 28–29  
ntdll.dll, **functions in**, 39  
ntoskrnl.exe, **kernel hooking problems and**, 42

## O

**obfuscate, defined**, 13  
oldFileSysDevice, **device pointer**, 146–150  
oldKeyboardDevice **global variable, key logging**,  
    **172–173**  
oldNetworkDevice, **device pointer**, 146–150  
OldZwMapViewOfSection  
    Ghost.c file variable, 33–36  
    hookManager.h file variable, 37–38  
OnCancel , **function**, 185  
OnCheckNames, **function**, 218–231  
OnCheckNamesComplete, **function**, 219–231  
OnDeviceControl, **function**, 210  
OnKeyboardRead, **function**, 184  
OnRead, **function**, 218–231  
OnReadComplete, **function**, 218–231  
OnReadCompletion, **function**, 184  
OnSendMail, **function**, 234–239  
OnSubmit, **function**, 216, 219–231  
OnSubmitComplete, **function**, 216, 219–231  
OnUnload, **function**, 172–173  
OnWrite, **function**, 218–231  
OnWriteComplete, **function**, 216, 218–231  
**opening, Local Kernel Debugger**, 7  
OpenTDIConnection, **function**, 122–130  
**operating system updates, rootkit prevention**, 292  
**outbound content compliance software. See**  
    **intended installation**  
**Outlook Client Extension testing, code**, 232  
OutlookExtension.cpp  
    code, 219–230  
    E-mail filtering implementation file, 216, 218–231  
    functions list, 218–219  
OutlookExtension.dsp, **E-mail filtering skeletal**  
    **file**, 216  
OutlookExtension.dsw, **E-mail filtering skeletal**  
    **file**, 216  
OutlookExtension.h  
    code, 216–218  
    E-mail filtering implementation file, 216–218

## P

parameters, CALL\_DATA\_STRUCT, 63  
parse86.c **file**  
    code, 79–96  
    functions list, 78–79  
parse86.h **file**  
    code, 78  
    functions list, 78  
ParseRecipientList, **function**, 234–239  
**parsing**  
    PE formatted files, 97–99  
    x86 instructions, 96  
**payload**  
    defined, 7  
    overview, 7–8  
**PE formatted files, parsing**, 97–99  
peFormat.h **file**  
    code, 97–99  
    user hooks, 97–99  
**periodic status reporting, feedback**, 244  
**persistence, installation**, 245–246  
**personal firewalls**  
    free, 294  
    to purchase, 294–295  
    rootkit prevention, 293–295  
**Pfx (ANSI Prefix Manager), functional group**, 40–41  
**PfxFindPrefix, routine**, 41  
**PfxInitialize, routine**, 40  
**PfxInsertPrefix, routine**, 41  
**PfxRemovePrefix, routine**, 40  
**PGP Desktop**  
    overview, 115–117  
    Professional version 9 download, 99  
**PGP encoding, using Ghost to block**, 99–100  
**PGP Monitor, Microsoft Windows 2000, XP, and**,  
    **2003**, 101  
**piggybacked, defined**, 289  
Ping, **function**, 269–270  
pMyMDL  
    Ghost.c file variable, 33–36  
    hookManager.h file variable, 37–38  
**Policy Development, control category**, 257  
**Policy Implementation, control category**, 257  
**Port operations, Zw routine**, 41  
**prevention. See rootkit prevention**  
**privilege escalation, overview**, 245  
**process creation detection, IceSword**, 314  
**Process detection, IceSword**, 313

**process hiding**

diagrammed, 206  
 HideMe.c file, 206–211  
 overview, 205–206  
 testing, 212

**process injection**

injectManager.c file and, 66–78  
 limitation of, 47  
 NewZwMapViewOfSection function, 47  
 overview, 43–44  
 trampoline function and, 49

**process injection hook, beforeEncode, 67–78****Process operations, Zw routine, 41****process termination detection, IceSword, 314****ProcessGuard, anti-rootkit software, 254****Processing exceptions, Rtl routine, 41****processing levels, key logging and, 167–168****processInject, function, 66–78****programming, injected function, 114****programs, compiling, 21, 23–24**

PsCreateSystemThread, **function, 170**

PsTerminateSystemThread, **function, 170**

PutFile, **function, 16–19, 20**

**Q**

QueryInterface, **function, 218–231, 219–231**

**R**

Readme.txt

E-mail filtering implementation file, 232

E-mail filtering skeletal file, 216

readme.txt **file**

code, 241

E-mail Filtering, 241

**Recipient Selection dialog box, 115–116**

RegisterEntry, **function, 234–239**

**Registry**

backing up the, 211–212

modification risks, 211

settings installation, 247–248

**registry key**

detecting, 276

Ghost.c file, 198

hookManager.c file, 199–202

hookManager.h file, 198–199

registryManager.c file, 189–198

registryManager.h file, 188–189

testing, 212

**Registry operations, Zw routine, 41****registry tamper detection, IceSword, 314**

registryManager.c **file**

code, 190–197

concealment, 189–198

functions list, 189–190

registryManager.h **file**

code, 188–189

concealment, 188–189

**RegistryMonitor**

FileMonitor Versus, 305

freeware, 302–304

**RegMon, utility, 2, 5–6****RegMon. See RegistryMonitor**

Release, **function, 218–231**

removeFilter, **function, 142–145**

**Reporting, control category, 257****resource functions, differentiated, 20****rootkit**

adding an on/off switch to the, 104–114

building overview, 1–3

comint32.sys, 21

creating a basic, 9–12

dealing with a detected, 287–289

detection methods, 275–279

detection summary, 290

device driver, 9–15

environment diagrammed, 134

installing a, 21–25

loading/unloading the, 24

summary, 26

testing a, 26

toolkit overview, 3

verifying the presence of a, 287

**rootkit controller**

the connection, 257

ControlForm, 273

ControlForm.cs file, 262–268

the controller, 255–257

example, 258–273

GhostTracker form, 273

GhostTracker.cs file, 260–262

Listen.cs file, 270–272

tamper detection, 257–258

TargetController.cs file, 268–270

## Rootkit Hook Analyzer

detection software, 282–283  
freeware, 311–312

### rootkit installation

SCMLoader.c, 22  
SCMUnloader.c, 25

### rootkit prevention

automatic updates, 292  
blocking unexpected operations, 298  
hardening, 295–297  
host-based intrusion prevention systems,  
295–298  
operating system updates, 292  
personal firewalls, 293–295  
summary, 299–300  
techniques, 298–299  
virtualizing, 297

**rootkit remote controller implementation,**  
**summary, 274**

**rootkit software, anti-, 254**

**rootkit tools, summary, 8**

### Rootkit Unhooker

freeware, 308–310  
software, 288

### RootkitRevealer

detection software, 280–281  
freeware, 310

**rootkits, preventing, 291–300**

**Rtl (Runtime Library), functional group, 41**

RtlInitUnicodeString, **definition of, 20**

**Runtime Library (Rtl), functional group, 41**

## S

**Safe Mode, entering, 289–290**

**sample, building a, 6**

**Samurai, freeware, 307–308**

**Samurai HIPS, hardening techniques, 296–297**

**Save As dialog box, 115–116**

**Save PGP Zip As dialog box, 115–116**

SaveAttachments, **function, 234–239**

SaveBody, **function, 234–239**

SaveRecipients, **function, 234–239**

**scanning, kernel memory, 278**

**Scheduling, control category, 257**

SCMLoader.c **file**

build environment problems, 23  
code, 22

Debug View output, 24  
VCVARS32.BAT file, 23

SCMUnloader.c **file**

build command, 25  
code, 25  
rootkit installation, 25

**semaphore guarded linked list, threading and  
synchronization technique, 170**

SendToRemoteController, **function, 122–130**

**server operations, functional groups for hooking, 39**

**Service Control Manager, ZwSetSystem  
Information, 246–247**

**service descriptor table, overview, 27–28**

**service detection, IceSword, 314**

**service load prevention, prevention technique, 298**

ServiceDescriptorEntry, hookManager.h **file,**  
**37–38**

**signature, defined, 248**

**software. See also detection software**

anti-rootkit, 254  
detection, 279–287  
InstallShield, 244, 287  
intended installation, 243–244  
MetaSploit, 8  
ProcessGuard, 167–168  
Strider GhostBuster, 280

### Sophos Anti-Rootkit

detection software, 286–287  
freeware, 315

### SOURCES

Basic Rootkit, 20  
Communications, 130–131  
Filter Drivers, 166  
Hooking the Kernel System Call Table, 33  
I/O Processing, 112  
Key Logging, 172  
User Hooks, 50

**SQL Server, integrating the, 5**

**stack execution prevention, prevention  
technique, 299**

stackOffset, CALL\_DATA\_STRUCT, **63**

Start, **function, 268–270, 270–272**

StartKeyLogger, **function, 174, 185**

Stdafx.cpp, **E-mail filtering skeletal file, 216**

Stdafx.h, **E-mail filtering skeletal file, 216**

Stop, **function, 269–270, 270–272**

StopKeyLogger, **function, 185**

**Strider GhostBuster, detection software, 280**

**string functions, differentiated, 20**

**summaries**

- Basic Rootkit, 26
- Communications, 135–136
- Concealment, 212–213
- E-mail Filtering, 242
- Filter Drivers, 166
- I/O Processing, 117–118
- Installation Considerations, 254
- Kernel Hooks, 42
- Key Logging, 186
- Rootkit Detection, 290, 299–300
- Rootkit Remote Controller Implementation, 274
- Rootkit Tools, 8
- Tools, 8
- User Hooks, 100–101

**Summary view, overview, 257**

**SwapContext**

- overview, 278
- Process Hiding Detection diagrammed, 279

**Sygate Personal Firewall, overview, 294**

**Symantec/Norton Firewall, overview, 295**

**symbols, downloading, 2–3**

**synchronization, functions list, 170**

**synchronization functions, differentiated, 20**

**Sysinternals**

- Freeware downloads, 5–6
- utilities, 2

**system call table**

- diagrammed, 31
- hooking diagrammed, 31
- hooking the, 30–31
- KeServiceDescriptorTable, 30
- trap checks of the, 42

**system service table, overview, 27–28**

## T

**tagging, tracked files with ADS, 277**

**tamper detection, rootkit controller, 257–258**

**TargetController, function, 268–270**

**TargetController.cs file**

- code, 269–270
- functions list, 268–269
- rootkit remote controller implementation, 268–270

**TargetController.mf, file, 268–270**

**targetListView\_SelectedIndexChanged, function, 260–262**

**TCP/IP connection, verifying compliance with a, 258**

**TCP, UDP, and RAW IP port activity detection, IceSword, 313**

**TCPView, freeware, 305**

**TDI (Transport Driver Interface)**

- demonstrating the, 133–135
- overview, 119–120

**TDICompletionRoutine, function, 122–130**

**techniques**

- for installing Mozilla Firefox, 249–251
- rootkit prevention, 298–299

**testing**

- concealment, 211–212
- file-hiding, 212
- I/O control, 114–117
- installation, 254
- the Lotus Notes client extension, 242
- the Outlook client extension, 231–232
- registry key, 212
- a rootkit, 26

**threading, functions list, 170**

**Timer operations**

- Rtl routine, 41
- Zw routine, 41

**TimerDPC, function, 122–130**

**Tiny Personal Firewall, overview, 294**

**Token operations, Zw routine, 41**

**tools**

- Debugging, 2, 7
- required for building a rootkit, 1–3
- summary, 8

**trace operations, functional groups for hooking, 40**

**trampoline, function, 48–49**

**trampoline**

- hooking detection, 277
- overview, 42
- process diagrammed, 49
- process and ZwMapViewOfSection, 49

**transferData, function, 78–96**

**transferDataPrefix, function, 78–96**

**transferInstruction, function, 78, 78–96**

**transferOp0F, function, 78–96**

**transferOp66, function, 78–96**

**transferOp67, function, 78–96**

**transferOpF6, function, 78–96**

# transferOpF7, function

---

**transferOpF7, function, 78–96**

**transferOpFF, function, 78–96**

## **Transport Driver Interface (TDI)**

demonstrating the, 133–135

overview, 119–120

## **U**

**UNHOOK, macro, 37–38**

### **Unicode string**

FileName, 20

specifier for a, 47

**unintended installation, overview, 245**

**unloading, the rootkit, 24**

**Updates, control category, 257**

### **User Hooks**

code for finding a specific dynamic link library, 44–46

example, 50–99

finding a specific library, 44–49

Ghost.h file, 50–51

Ghost.h file code, 51

hookManager.c file, 54–63

hookManager.c file code, 55–63

hookManager.h file, 52–54

hookManager.h file code, 52–54

injectManager.c file, 66–78

injectManager.c file code, 67–78

injectManager.h file, 63–66

injectManager.h file code, 63–66

parse86.c file, 78–96

parse86.c file code, 79–96

parse86.h file, 78

parse86.h file code, 78

peFormat.h file, 97–99

peFormat.h file code, 97–99

process injection, 43–44

SOURCES, 50

summary, 100–101

using Ghost to block PGP encoding, 99–100

## **V**

**VCVARS32.BAT file**

Command Prompt window, 23

using, 7

**verification, installation, 6–7**

### **verifying**

Debugging Tools, 2, 7

Debugging Tools for Windows, 2

Microsoft Driver Development Kit (DDK), 6

Microsoft Visual C++ 2005 Express, 6

rootkit presence, 287

**virtualizing, overview, 297**

**Visual C# compiler, downloading, 3**

## **W**

### **website**

Microsoft, 1–2

Wrox, 8

**Windows Firewall dialog box, 293**

**Windows Platform Software Development Kit (SDK)**

downloading the, 1–2

installing the, 5

**Windows Server 2003 SP1 DDK CD, ordering the, 2**

**Windows XP personal firewall, using the, 293**

**winsock catalog entry detection, IceSword, 314**

**Wrox, website, 8**

## **Z**

**Zone Alarm Firewall, overview, 294**

**Zone Alarm Professional Firewall, overview, 295**

**Zw (File and Registry), functional group, 41**

ZwMapViewOfSection

diagrammed, 44

Ghost.c, 34

hooking, 33

process injection, 43–44

trampoline process and, 49

ZwOpenFile **kernel mode device driver, 20**

ZwProtectVirtualMemory **variable, Ghost.c**

**file, 51–52**

ZwSetSystemInformation, **Service Control**

**Manager, 246–247**







