Inside Windows Debugging A Practical Guide To Debugging And Tracing Strategies In Windowsi 1 2 I 1 2 By Tarik Soulami 24 May 2012 Paperback

JIMMY GINA

Introducing Windows 10 for IT Professionals "O'Reilly Media, Inc."
A reference book for technical support and escalation engineers troubleshooting and debugging complex software issues. The book is also invaluable for software maintenance and development engineers debugging Windows applications and services.
Practical Foundations of Windows Debugging, Disassembling, Reversing No Starch Press
See how the core components of the Windows operating system work behind the scenes—guided by a team of internationally renowned internals experts. Fully updated for Windows Server(R) 2008 and Windows Vista(R), this classic guide delivers key architectural insights on system design, debugging, performance, and support—along with hands-on experiments to experience Windows internal behavior firsthand. Delve inside Windows architecture and internals: Understand how the core system and management mechanisms work—from the object manager to services to the registry Explore internal system data structures using tools like the kernel debugger Grasp the scheduler's priority and CPU placement algorithms Go inside the Windows security model to see how it authorizes access to data Understand how Windows manages physical and virtual memory Tour the Windows networking stack from top to bottom—including APIs, protocol drivers, and network adapter drivers Troubleshoot file-system access problems and system boot problems Learn how to analyze crashes
Practical Debugging for . NET Developers Pearson Education
This training course is a combined, reformatted, improved, and modernized version of the two previous books (x64) Windows Debugging: Practical Foundations, that drew inspiration from the original lectures we developed almost 18 years ago to train support and escalation engineers in debugging and crash dump analysis of memory dumps from Windows applications, services, and systems. At that time, when thinking about what material to deliver, we realized that a solid understanding of fundamentals like pointers is needed to analyze stack traces beyond a few WinDbg commands. Therefore, this book is not about bugs or debugging techniques but about the background knowledge everyone needs to start experimenting with WinDbg and learn from practical experience and read other advanced debugging books. This body of knowledge is what the author of this book possessed before starting memory dump analysis using WinDbg 18 years ago, which resulted in the number one debugging bestseller: multi-volume Memory Dump Analysis Anthology. Now, in retrospection, we see these practical foundations as relevant and necessary to acquire for beginners as they were 18 years ago because operating systems internals, assembly

language, and compiler architecture haven't changed much in those years. The book contains two separate sets of chapters and corresponding illustrations. They are named Chapter x86.NN and Chapter x64.NN respectively. The new format makes switching between and comparing x86 and x64 versions easy. Both sets of chapters can be read independently. We included x86 chapters because many 3rd-party Windows applications are still 32-bit and executed in 32-bit compatibility mode on x64 Windows systems. Almost 5 years have passed since the first edition of the combined training course that used the earlier version of Windows 10. Since then, we have also published "Practical Foundations of Linux Debugging, Disassembling, Reversing" and "Practical Foundations of ARM64 Linux Debugging, Disassembling, Reversing" books. At that time, we thought about revising our Windows course. Since then, Windows 11 appeared, and we also added Docker support for most of our Windows memory dump analysis courses. While working on the "Accelerated Windows Debugging 4D "course, we decided to make the second edition of Practical Foundations of Windows Debugging based on WinDbg from Windows 11 SDK and Visual Studio 2022 build tools and an optional Docker support for the exercise environment. We also changed the " =" operator to "" in pseudo-code for x64 AT&T disassembly syntax flavor and " The book is useful for: - Software technical support and escalation engineers; - Software engineers coming from managed code or JVM background; - Software testers; - Engineers coming from non-Wintel environments; - Windows C/C++ software engineers without assembly language background; - Security researchers without x86/x64 assembly language background; - Beginners learning Windows software reverse engineering techniques; This introductory training course can complement the more advanced course Accelerated Disassembly, Reconstruction and Reversing, Revised Edition. It may also help with advanced exercises in Accelerated Windows Memory Dump Analysis books. This book can also be used as an Intel assembly language and Windows debugging supplement for relevant undergraduate-level courses.

Debugging Applications for Microsoft .NET and Microsoft Windows Microsoft Press

With the growing prevalence of the Internet, rootkit technology has taken center stage in the battle between White Hats and Black Hats. Adopting an approach that favors full disclosure, The Rootkit Arsenal presents the most accessible, timely, and complete coverage of rootkit technology. This book covers more topics, in greater depth, than any other currently available. In doing so, the author forges through the murky back alleys of the Internet, shedding light on material that has traditionally been poorly documented, partially documented, or intentionally undocumented.

Debugging Microsoft .NET 2.0 Applications Fastprint Publishing

Debugging is crucial to successful software development, but even many experienced programmers find it challenging. Sophisticated debugging tools are available, yet it may be difficult to determine which features are useful in which situations. The Art of Debugging is your guide to making the debugging process more efficient and effective. The Art of Debugging illustrates the use three of the most popular debugging tools on Linux/Unix platforms: GDB, DDD, and Eclipse. The text-command based GDB (the GNU Project Debugger) is included with most distributions. DDD is a popular GUI front end for GDB, while Eclipse provides a complete integrated development environment. In addition to offering specific advice for debugging with each tool, authors Norm Matloff and Pete Salzman cover general strategies for improving the process of finding and fixing coding errors, including how to: –Inspect variables and data structures –Understand segmentation faults and core dumps –Know why your program crashes or throws exceptions –Use features like catchpoints, convenience variables, and artificial arrays –Avoid common debugging pitfalls Real world examples of coding errors help to clarify the authors' guiding principles, and coverage of complex topics like thread, client-server, GUI, and parallel programming debugging will make you even more proficient. You'll also learn how to prevent errors in the first place with text editors, compilers, error reporting, and static code checkers. Whether you dread the thought of debugging your programs or simply want to improve your current debugging efforts, you'll find a valuable ally in The Art of Debugging.

Inside Windows Debugging Wiley

While forensic analysis has proven to be a valuable investigative tool in the field of computer security, utilizing anti-forensic technology makes it possible to maintain a covert operational foothold for extended periods, even in a high-security environment. Adopting an approach that favors full disclosure, the updated Second Edition of The Rootkit Arsenal presents the most accessible, timely, and complete coverage of forensic countermeasures. This book covers more topics, in greater depth, than any other currently available. In doing so the author forges through the murky back alleys of the Internet, shedding light on material that has traditionally been poorly documented, partially documented, or intentionally undocumented. The range of topics presented includes how to: -Evade post-mortem analysis -Frustrate attempts to reverse engineer your command & control modules -Defeat live incident response -Undermine the process of memory analysis -Modify subsystem internals to feed misinformation to the outside -Entrench your code in fortified regions of execution -Design and implement covert channels -Unearth new avenues of attack

Hacker Debugging Uncovered Addison-Wesley Professional

Delve inside Windows architecture and internals—and see how core components work behind the scenes. Led by three renowned internals experts, this classic guide is fully updated for Windows 7 and Windows Server 2008 R2—and now presents its coverage in two volumes. As always, you get critical insider perspectives on how Windows operates. And through hands-on experiments, you'll experience its internal behavior firsthand—knowledge you can apply to improve application design, debugging, system performance, and support. In Part 1, you will: Understand how core system and management mechanisms work—including the object manager, synchronization, Wow64, Hyper-V, and the registry Examine the data structures and activities behind processes, threads, and jobs Go inside the Windows security model to see how it manages access, auditing, and authorization Explore the Windows networking stack from top to bottom—including APIs, BranchCache, protocol and NDIS drivers, and layered services Dig into internals hands-on using the kernel debugger, performance monitor, and other tools

Rootkit Arsenal McGraw-Hill Companies

This book doesn't tell you how to write faster code, or how to write code with fewer memory leaks, or even how to debug code at all. What it does tell you is how to build your product in better ways, how to keep track of the code that you write, and how to track the bugs in your code. Plus some more things you'll wish you had known before starting a project. Practical Development Environments is a guide, a collection of advice about real development environments for small to medium-sized projects and groups. Each of the chapters considers a different kind of tool - tools for tracking versions of files, build tools, testing tools, bug-tracking tools, tools for creating documentation, and tools for creating packaged releases. Each chapter discusses what you should look for in that kind of tool and what to avoid, and also describes some good ideas, bad ideas, and annoying experiences for each area. Specific instances of each type of tool are described in enough detail so that you can decide which ones you want to investigate further. Developers want to write code, not maintain makefiles. Writers want to write content instead of manage templates. IT provides machines, but doesn't have time to maintain all the different tools. Managers want the product to move smoothly from development to release, and are interested in tools to help this happen more often. Whether as a full-time position or just because they are helpful, all projects have toolsmiths: making choices about tools, installing them, and then maintaining the tools that everyone else depends upon. This book is especially for everyone who ends up being a toolsmith for his or her group.

Advanced .NET Debugging БХВ-Петербург

Get a head start evaluating Windows 10--with technical insights from award-winning journalist and Windows expert Ed Bott. This guide introduces new features and capabilities, providing a practical, high-level overview for IT professionals ready to begin deployment planning now. This edition was written after the release of Windows 10 version 1511 in November 2015 and includes all of its enterprise-focused features. The goal of this book is to help you sort out what's new in Windows 10, with a special emphasis on features that are different from the Windows versions you and your organization are using today, starting with an overview of the operating system, describing the many changes to the user experience, and diving deep into deployment and management tools where it's necessary.

Practical Foundations of Windows Debugging, Disassembling, Reversing Pearson Education

Tips for the practical use of debuggers, such as NuMega SoftIce, Microsoft Visual Studio Debugger, and Microsoft Kernel Debugger, with minimum binding to a specific environment are disclosed in this debugger guide. How debuggers operate and how to overcome obstacles and repair debuggers is demonstrated. Programmers will learn how to look at what is inside a computer system, how to reconstruct the operating algorithm of a program distributed without source code, how to modify the program, and how to debug drivers. The use of debugging applications and drivers in Windows and Unix operating systems on Intel Pentium/DEC Alpha-based processors is also detailed.

Mastering Visual Studio .NET Microsoft Press

A total guide to debuggers: what they do, how they work, and how to use them to produce better programs "Debuggers are the magnifying glass, the microscope, the logic analyzer, the profiler, and the browser with which a program can be examined."-Jonathan B. Rosenberg Debuggers are an indispensable tool in the development process. In fact, during the course of the average software project, more hours are spent debugging software than in compiling code. Yet, not many programmers really know how to constructively interpret the results they get back from debuggers. And even fewer know what makes these complex suites of algorithms and data structures tick. Now in this extremely accessible guide, Jonathan B. Rosenberg demystifies debuggers for programmers and shows them how to make better use of debuggers in their next projects. Taking a hands-on, problem-solving approach to a complex subject, Rosenberg explains how debuggers work and why programmers use them. Most importantly, he provides practical discussions of debugger algorithms and procedures for their use, accompanied by many practical examples. The author also discusses a wide variety of systems applications, from Microsoft's Win32 debug API to a large parallel architecture. Visit our Web site at: http://www.wiley.com/compbooks/

Debugging Windows Programs HarperChristian + ORM

"Jocelyn Brooke is a great writer. . . . If you care enough for literature, seek out The Scapegoat."--Elizabeth Bowen "Brooke marked out his magical, personal kingdom, different from any other writer."--Anthony Powell

Windows 10 Inside Out (includes Current Book Service) Addison-Wesley Professional

Use Windows debuggers throughout the development cycle—and build better software Rethink your use of Windows debugging and tracing tools—and learn how to make them a key part of test-driven software development. Led by a member of the Windows Fundamentals Team at Microsoft, you'll apply expert debugging and tracing techniques—and sharpen your C++ and C# code analysis skills—through practical examples and common scenarios. Learn why experienced developers use debuggers in every step of the development process, and not just when bugs appear. Discover how to: Go behind the scenes to examine how powerful Windows debuggers work Catch bugs early in the development cycle with static and runtime analysis tools Gain practical strategies to tackle the most common code defects Apply expert tricks to handle user-mode and kernel-mode debugging tasks Implement postmortem techniques such as JIT and dump debugging Debug the concurrency and security aspects of your software Use debuggers to analyze interactions between your code and the operating system Analyze software behavior with Xperf and the Event Tracing for Windows (ETW) framework

Perl Debugged No Starch Press

Every software developer and IT professional understands the crucial importance of effective debugging. Often, debugging consumes most of a developer's workday, and mastering the required techniques and skills can take a lifetime. In Effective Debugging, Diomidis Spinellis helps experienced programmers accelerate their journey to mastery, by systematically categorizing, explaining, and illustrating the most useful debugging methods, strategies, techniques, and tools. Drawing on more than thirty-five years of experience, Spinellis expands your arsenal of debugging techniques, helping you choose the best approaches for each challenge. He presents vendor-neutral, example-rich advice on general principles, high-level strategies, concrete techniques, high-efficiency tools, creative tricks, and the behavioral traits associated with effective debugging. Spinellis's 66 expert techniques address every facet of debugging and are illustrated with step-by-step instructions and actual code. He addresses the full spectrum of problems that can arise in modern software systems, especially problems caused by complex interactions among components and services running on hosts scattered around the planet. Whether you're debugging isolated runtime errors or catastrophic enterprise system failures, this guide will help you get the job done—more quickly, and with less pain. Key features include High-level strategies and methods for addressing diverse software failures Specific techniques to apply when programming, compiling, and running code Better ways to make the most of your debugger General-purpose skills and tools worth investing in Advanced ideas and techniques for escaping dead-ends and the maze of complexity Advice for making programs easier to debug Specialized approaches for debugging multithreaded, asynchronous, and embedded code Bug avoidance through improved software design, construction, and management

Windows Internals, Part 1 Pearson Education

Analyzing how hacks are done, so as to stop them in the future Reverse engineering is the process of analyzing hardware or software and understanding it, without having access to the source code

or design documents. Hackers are able to reverse engineer systems and exploit what they find with scary results. Now the good guys can use the same tools to thwart these threats. Practical Reverse Engineering goes under the hood of reverse engineering for security analysts, security engineers, and system programmers, so they can learn how to use these same processes to stop hackers in their tracks. The book covers x86, x64, and ARM (the first book to cover all three); Windows kernel-mode code rootkits and drivers; virtual machine protection techniques; and much more. Best of all, it offers a systematic approach to the material, with plenty of hands-on exercises and real-world examples. Offers a systematic approach to understanding reverse engineering, with hands-on exercises and real-world examples Covers x86, x64, and advanced RISC machine (ARM) architectures as well as deobfuscation and virtual machine protection techniques Provides special coverage of Windows kernel-mode code (rootkits/drivers), a topic not often covered elsewhere, and explains how to analyze drivers step by step Demystifies topics that have a steep learning curve Includes a bonus chapter on reverse engineering tools Practical Reverse Engineering: Using x86, x64, ARM, Windows Kernel, and Reversing Tools provides crucial, up-to-date guidance for a broad range of IT professionals.

Practical Reverse Engineering "O'Reilly Media, Inc."

The full transcript of Software Diagnostics Services training with step-by-step exercises, notes, and source code to learn live local and remote debugging techniques in kernel, user process and managed .NET spaces using WinDbg debugger. The second edition was fully reworked and updated to use the latest WinDbg version and Windows 10.

Debugging "O'Reilly Media, Inc."

The definitive guide–fully updated for Windows 10 and Windows Server 2016 Delve inside Windows architecture and internals, and see how core components work behind the scenes. Led by a team of internals experts, this classic guide has been fully updated for Windows 10 and Windows Server 2016. Whether you are a developer or an IT professional, you'll get critical, insider perspectives on how Windows operates. And through hands-on experiments, you'll experience its internal behavior firsthand–knowledge you can apply to improve application design, debugging, system performance, and support. This book will help you: · Understand the Window system architecture and its most important entities, such as processes and threads · Examine how processes manage resources and threads scheduled for execution inside processes · Observe how Windows manages virtual and physical memory · Dig into the Windows I/O system and see how device drivers work and integrate with the rest of the system · Go inside the Windows security model to see how it manages access, auditing, and authorization, and learn about the new mechanisms in Windows 10 and Server 2016

Practical Malware Analysis Microsoft Press

For professional software developers, debugging is a way of life. This book is the definitive guide to Windows debugging, providing developers with the strategies and techniques they need to fulfill one of their most important responsibilities efficiently and effectively. Debugging Windows Programs shows readers how to prevent bugs by taking full advantage of the Visual C++ development tools and writing code in a way that makes certain types of bugs impossible. They also will learn how to reveal bugs with debugging statements that force bugs to expose themselves when the program is executed, and how to make the most of debugging tools and features available in Windows, Visual C++, MFC, and ATL. The authors provide specific solutions to the most common debugging problems, including memory corruption, resource leaks, stack problems, release build problems, finding crash locations, and multithreading problems. These essential topics are covered: The debugging process Writing C++ code for debugging Strategically using assertions, trace statements, and exceptions Windows postmortem debugging using Dr. Watson and MAP files Using the Visual C++ debugger Debugging memory Debugging multithreaded programs Debugging COM Each chapter provides developers with exactly what they need to master the subject and improve development productivity and software quality. Comprehensive, current, and practical, Debugging Windows Programs helps developers understand the debugging process and make the most of the Visual C++ debugging tools. 020170238XB04062001

Advanced .NET Debugging Pearson Education

"Raymond Chen is the original raconteur of Windows." --Scott Hanselman, ComputerZen.com "Raymond has been at Microsoft for many years and has seen many nuances of Windows that others could only ever hope to get a glimpse of. With this book, Raymond shares his knowledge, experience, and anecdotal stories, allowing all of us to get a better understanding of the operating system that affects millions of people every day. This book has something for everyone, is a casual read, and I highly recommend it!" --Jeffrey Richter, Author/Consultant, Cofounder of Wintellect "Very interesting read. Raymond tells the inside story of why Windows is the way it is." --Eric Gunnerson, Program Manager, Microsoft Corporation "Absolutely essential reading for understanding the history of Windows, its intricacies and quirks, and why they came about." --Matt Pietrek, MSDN Magazine's Under the Hood Columnist "Raymond Chen has become something of a legend in the software industry, and in this book you'll discover why. From his high-level reminiscences on the design of the Windows Start button to his low-level discussions of GlobalAlloc that only your inner-geek could love, The Old New Thing is a captivating collection of anecdotes that will help you to truly appreciate the difficulty inherent in designing and writing quality software." --Stephen Toub, Technical Editor, MSDN Magazine Why does Windows work the way it does? Why is Shut Down on the Start menu? (And why is there a Start button, anyway?) How can I tap into the dialog loop? Why does the GetWindowText function behave so strangely? Why are registry files called "hives"? Many of Windows' quirks have perfectly logical explanations, rooted in history. Understand them, and you'll be more productive and a lot less frustrated. Raymond Chen--who's spent more than a decade on Microsoft's Windows development team--reveals the "hidden Windows" you need to know. Chen's engaging style, deep insight, and thoughtful humor have made him one of the world's premier technology bloggers. Here he brings together behind-the-scenes explanations, invaluable technical advice, and illuminating anecdotes that bring Windows to life--and help you make the most of it. A few of the things you'll find inside: What vending machines can teach you about effective user interfaces A deeper understanding of window and dialog management Why performance optimization can be so counterintuitive A peek at the underbelly of COM objects and the Visual C++ compiler Key details about backwards compatibility--what Windows does and why Windows program security holes most developers don't know about How to make your program a better Windows citizen

Windows Debugging Jones & Bartlett Publishers

"Mario Hewardt's Advanced .NET Debugging is an excellent resource for both beginner and experienced developers working with .NET. The book is also packed with many debugging tips and discussions of CLR internals, which will benefit developers architecting software." –Jeffrey Richter, consultant, trainer, and author at Wintellect "Mario has done it again. His Advanced Windows Debugging (coauthored with Daniel Pravat) is an invaluable resource for native code debugging, and Advanced .NET Debugging achieves the same quality, clarity, and breadth to make it just as invaluable for .NET debugging." –Mark Russinovich, Technical Fellow, Microsoft Corporation The Only Complete, Practical Guide to Fixing the Toughest .NET Bugs Advanced .NET Debugging is the first focused, pragmatic guide to tracking down today's most complex and challenging .NET application bugs. It is the only book to focus entirely on using powerful native debugging tools, including WinDBG, NTSD, and CDB, to debug .NET applications. Using these tools, author Mario Hewardt explains how to identify the real root causes of problems—far more quickly than you ever could with other debuggers. Hewardt first introduces the key concepts needed to successfully use .NET's native debuggers. Next, he turns to sophisticated debugging techniques, using real-world examples that demonstrate many common C# programming errors. This book enables you to Make practical use of postmortem debugging, including PowerDBG and other "power tools" Understand the debugging details and implications of the new .NET CLR 4.0 Master and successfully use Debugging Tools for Windows, as well as SOS, SOSEX, CLR Profiler, and other powerful tools Gain a deeper, more practical understanding of CLR internals, such as examining thread-specific data, managed heap and garbage collector, interoperability layer, and .NET exceptions Solve difficult synchronization problems, managed heap problems, interoperability problems, and much more Generate and successfully analyze crash dumps