Dmitry Vostokov

Pattern-Oriented Software Diagnostics, Forensics, Prognostics, Root Cause Analysis, Debugging. Diagnostics of Things.

dmitry.vostokov@patterndiagnostics.com

Summary

Dmitry Vostokov is an internationally recognized expert, speaker, educator, scientist and author. He is the founder of pattern-oriented software diagnostics, forensics and prognostics discipline and Software Diagnostics Institute (DA+TA: DumpAnalysis.org + TraceAnalysis.org). Vostokov has also authored more than 30 books on software diagnostics, forensics and problem solving, memory dump analysis, debugging, software trace and log analysis, reverse engineering and malware analysis. He has more than 20 years of experience in software architecture, design, development and maintenance in a variety of industries including leadership, technical and people management roles. Dmitry also founded DiaThings, Logtellect, OpenTask Iterative and Incremental Publishing (OpenTask.com), Software Diagnostics Services (former Memory Dump Analysis Services) PatternDiagnostics.com and Software Prognostics. In his spare time, he presents various topics on Debugging.TV and explores Software Narratology, an applied science of software stories that he pioneered, and its further development as Narratology of Things and Diagnostics of Things (DoT). His current area of interest is theoretical software diagnostics.

Timestamp: 3 April 2017

Experience

Software Support Visionary at Citrix

October 2003 - Present (13 years 7 months)

Titles included: Senior Software Engineer, Principal Dev Analysis Engineer, Technical Manager Dev Analysis EMEA, EMEA Dev Analysis Team Lead, Escalation Dev Analysis Engineer. While performing in various software support, engineering, and people management roles generated ideas behind many company projects. The description is currently under NDA.

Founder at DiaThings

April 2016 - Present (1 year 1 month)

A subsidiary of Software Diagnostics Services. Develops pattern-oriented methodology, solutions, training and certification for Diagnostics of Things (DoT) based on pattern-oriented diagnostics and Narratology of Things.

Founder and CEO at Logtellect

September 2015 - Present (1 year 8 months)

Logtellect, a subsidiary of Software Diagnostics Services, develops trace and log analysis products, solutions, and training courses.

Founder, President and CEO at Software Diagnostics Services

July 2010 - Present (6 years 10 months)

Software Diagnostics Services (former Memory Dump Analysis Services) provides subscription, incident-based abnormal software behaviour analysis support for enterprise customers around the world. Develops training courses in software diagnostics and debugging, malware analysis, memory dump analysis and memory forensics, reversing, software trace and log analysis.

Founder and Chief Scientist at Software Diagnostics Institute

March 2006 - Present (11 years 2 months)

Structural and Behavioural Patterns for Software Diagnostics, Forensics and Prognostics. Software Diagnostics Library.

Founder and Editor-in-Chief at OpenTask Publishing

February 2008 - Present (9 years 3 months)

Iterative and Incremental Publishing. Technical book design and production.

Founder and Presenter at Debugging TV

October 2011 - Present (5 years 7 months)

Broadcasts TV series called Frames where each episode features some facet of debugging, memory dump or software trace and log analysis including live WinDbg (Windows) or GDB demonstration (Mac OS X, Linux).

Founder and CEO at Software Prognostics

December 2013 - Present (3 years 5 months)

A subsidiary of Software Diagnostics Services. Originally called Zero Fault Software Diagnostics (fault prognosis). Further develops pattern-oriented software diagnostics to anticipate faults before their occurrence.

Founder at Tool Objects

October 2010 - Present (6 years 7 months)

A subsidiary of Software Diagnostics Services. Develops innovative solutions for the complexity of software diagnostics, troubleshooting and debugging.

Founder and Chief Diagnostics Scientist at Diagnostics Science

March 2013 - Present (4 years 2 months)

All areas of human activity involve the use of diagnostics. Proper diagnostics identifies the right problems to solve. A non-profit organization dedicated to the developing and promoting the application of systemic and pattern-oriented (pattern-driven and pattern-based) diagnostics.

Founder and Editor-in-Chief at Debugging Experts Magazine Online (Debugged! MZ/PE)
March 2009 - Present (8 years 2 months)
An online and print magazine about software debugging.

Software Development Consultant at Guardeonic Solutions
April 2003 - September 2003 (6 months)
Used: Windows NT/2000/XP/2003, WinDbg, C, C++, SQL, UML, XML, XSD, Visio, Poseidon, CVS, Visual C++, MFC, COM/DCOM, STL, PKI, ASN.1, Cryptography (DES), Win32 API, Platform SDK, Windows NT/2000/XP DDK, GINA, Certificates, smartcard and fingerprint technology.

Senior Software Engineer at Programming Research
September 2001 - March 2003 (1 year 7 months)
- Responsible for feasibility studies, requirements analysis, architecture, design and implementation of several subsystems of deep flow static analysis tools for C and C++.
- Responsible for technology evaluation and research (semantics of .NET languages, C++, templates and STL).

OS Platforms: Windows 2000, Unix (Linux)
Languages: C, C99, C++, Embedded C++, UML, OCL, XML
Tools : Microsoft Visual Modeler, CVS, Wiki, VMware, GNU C++, Comeau C++, Intel C++, Visual C++, Visual C++.NET, QAC, QAC++, STL, Boost, eXtreme Programming
Technologies: Compiler front ends, scanners, parsers, data flow analysis, metrics, garbage collection, virtual machines, CLR, .NET

- Designed and implemented a high-performance multiplatform object-relational database for a compiler front end. Responsible for feasibility studies, requirements, architecture and design documentation.

OS Platforms: Windows 2000, Unix (Linux)
Languages: C++, UML, OCL, XML, SQL
Tools: Microsoft Visual Modeler, CVS, Wiki, VMware, GNU C++, Visual C++, QAC++, STL, eXtreme Programming
Technologies: Client-server, ORDBMS

- Designed and implemented Adobe Acrobat plug-in for ISO C++ Standard documentation.

OS Platforms: Windows 2000, Unix (Linux)
Languages: C
Tools: CVS, VMware, GNU C++, Visual C++, Adobe Acrobat SDK, eXtreme Programming

Senior Software Designer at Ericsson

January 2001 - August 2001 (8 months)

Member of an engineering team responsible for analysis, design and implementation of core operation and maintenance interfaces for WCDMA/UMTS Media Gateway Cello platform. Responsible for use case design realization and implementation in C++. Created about 100 sequence and statechart UML diagrams. Responsible for portability across different platforms and languages. Debugged Rational Rose RealTime implementation models. Extended subsystems to support distributed clusters. Mentored 4 software engineers.

OS Platforms: Windows NT, Unix (Solaris), OSE Delta
Languages: C, C++, Java, SQL, UML, ROOM, SDL
Tools: Rational Rose RealTime, Rational Unified Process, ClearCase, Visual C++, ObjecTime, Microsoft Project
Technologies: Multi-tier, real time distributed systems, clusters, transactions, in-memory RDBMS, CORBA

Senior Software Engineer at Luxoft

July 2000 - January 2001 (7 months)

Former part of IBS Group where I worked the previous 9 months. ISO 9001 and CMM Level 4 Certified Company.

- Project lead and architect of a multiplatform enterprise PDF manipulation, linearization and rasterization system for Boeing Commercial Airplanes Group. Designed and implemented its architectural prototype in C++. Responsible for feasibility, requirements and design documentation, system integration and deployment. Mentored 6 software engineers.

OS Platforms: Windows NT, Unix (Linux, Solaris)
Languages: C, C++, XML, DTD, PDF, UML, OCL
Tools: Visio, Rational Rose, Rational SoDA, Rational Unified Process, VSS, VMware, Visual C++, STL, GNU C++, Microsoft Project
Technologies: SAX XML parsers, regular expression parsers, CORBA (ACE/TAO)

- Member of a team responsible for a web portal design and implementation for a large telecommunication company. Designed several subsystems. Responsible for design documentation.

Languages: Java 2 Enterprise Edition, UML, XML, XSL, HTML, WML
Tools: Rational Rose, Rational Unified Process, VSS, BEA Weblogic Application server
Technologies: Multi-tier, Servlets, WAP, DOM XML parsers, XSLT

Application Development Consultant at Accusoft Pegasus

April 1998 - December 2000 (2 years 9 months)

Remote work for a USA company located in Boston. Designed, implemented, and maintained the following projects:

- A wrapper to Scansoft TextBridge OCR API and ActiveX control. Wrote about 5,000 lines in C, 4,000 lines in C++ and 2,000 lines in Visual Basic.
- A Wireless BMP image converter to and from an internal image representation of a document imaging toolkit. Wrote about 1,000 lines in C.
- An ATL component for Image Annotation Toolkit. Wrote about 8,000 lines in C++.
- An image manipulation GUI library. Designed and implemented an ATL component. Designed and coded about 40 dialogs, wrote about 10,000 lines in C, 3,000 lines in C++ and 1,000 lines in Visual Basic.
- A JavaBean for Java Document Imaging Toolkit. Wrote about 13,000 lines in Java 1.1.
- Java Image Annotation classes. Wrote about 2,500 lines in Java 1.1.
- A multithreaded PDF rasterization extension for Document Imaging Toolkit.
- Member of an engineering team responsible for design and development of Java Document Imaging Toolkit (about 60,000 lines in Java). Implemented Java classes for image I/O via TCP/IP sockets. Wrote and tested about 3,000 lines in Java 1.1.
- Maintained and extended Win32 DLL and ActiveX MFC components for Image Annotation Toolkit as a part of Document Imaging Toolkit (more than 1,000,000 lines in C and C++). Responsible for API documentation, refactoring, extending functionality, testing and communication with technical support. Refactored about 50,000 lines in C and 20,000 lines in C++.
- Maintained Bar code recognition component.

Languages: C, C++, Visual Basic, VBScript, PDF, Java
Tools: VSS, Visual C++, Visual Basic, MFC, ATL, Win32 SDK, Borland JBuilder, Visual J++, Symantec Visual Cafe, Ghostscript, Delphi, MS Access, Microsoft Project
Technologies: COM, ActiveX, Client-server, TCP/IP, sockets, JavaBeans

Senior Software Engineer at IBS Group
October 1999 - June 2000 (9 months)

- Member of a team responsible for a web portal design and implementation for a large web hosting company. Responsible for technology evaluation, feasibility, requirements and vision documentation.

OS Platforms: Windows NT, Unix (Solaris)
Languages: Java 2 Enterprise Edition, UML, HTML, XML, XSLT
Tools: Rational Rose, Rational Unified Process, VSS, Sybase PowerDesigner/DataArchitect, BEA Weblogic Application and Personalization servers, Oracle 8.x
Technologies: Multi-tier, EJB, Servlets, JSP, DOM XML parsers

- Project lead and architect of Windows SGML Table validation and editing system for Boeing Commercial Airplanes Group. Designed and implemented GUI and system-level subsystems. Responsible for feasibility, requirements and design documentation. Wrote about 10,000 lines in Visual Basic and 1,000 lines in C. Maintained SGML to RTF and RTF to SGML converters written in C++. Responsible for system integration and deployment.

Languages: C, C++, Visual Basic, SGML, RTF, UML
Tools: Rational Rose, Rational Unified Process, VSS, Visio, Visual C++, STL, Visual Basic, Win32 SDK, Microsoft Project
Technologies: COM, OLE Automation

- Member of an engineering team responsible for C++ corporate software coding standard.
Responsible for Visual Basic corporate software coding standard.

Senior System Software Engineer at Interactive Products
March 1994 - August 1999 (5 years 6 months)

Remote work for a USA company Interactive Products, Inc. founded by former Covox employees and located in Oregon. Designed and implemented the following projects:

- A secure wrapper for Windows screen savers and a secure electronic communication system for Windows NT and 9x that used instant verbal confirmation of a person's identity. Wrote about 15,000 lines in C++ and 1,000 lines in x86 assembler.
- Personal notebooks for Windows with ability to store compressed speech. Wrote about 6,000 lines in C++.
- A word processor DLL with word-wrapped pictures, object embedding and events. Wrote about 20,000 lines in C++.
- OpenTask - a visual prototyping system for Windows designed to carry out any task and visually create prototypes, interactive presentations, tutorials, instructions and educational programs using build-in compiler. Wrote about 10,000 lines in C++.
- A system for handicapped people which controls mouse by voice. Wrote about 1,000 lines in C.
- A speech recognition system which enabled voice control of any software application on Windows 95/NT. Designed and implemented a special version which enabled voice control of CAD/CAM systems. Localized the products for Spanish language. Designed and implemented a speech recognition server for Windows which facilitated client applications to use voice recognition. Wrote about 45,000 lines in C++ and 1,500 lines in x86 assembler.
- A speech recognition system which enabled voice control of any software application on Windows 3.x. Localized the product for Japanese language (NEC 9800 platform). Wrote about 20,000 lines in C++ and 1,000 lines in x86 assembler.
- Win16 API emulation library for text-based MS-DOS applications. Wrote about 8,000 lines in C and 3,000 lines in x86 assembler.

Languages: C, C++, x86 assembly language
Tools: Borland C++, TASM, Visual C++, Windows DDK and SDK, MASM, VSS, STL, MFC
Technologies: COM, Win32 CryptoAPI, OLE Automation, Client-server, DDE, CORBA

Application Development Consultant at PC Voice Club
June 1994 - December 1998 (4 years 7 months)

- Designed and implemented an image viewer application. Wrote about 2,000 lines in Visual Basic.
- Designed and implemented a simple relational database and an applet for browsing that database. Wrote about 2,500 lines in Java 1.1 and 500 lines in Perl.
- Designed a bilingual (Russian/English) text-to-speech system with embedded multimedia and presentation capabilities. Implemented GUI, presentation end editing components. Wrote about 14,000 lines in C++.
- Designed and implemented a script language editor with syntax highlighting and word-processing capabilities. Wrote about 10, 000 lines in Java 1.0.
- Designed and implemented several games in Java (client and server parts). Wrote about 8,000 lines in Java 1.0. Designed a protocol for communication between server and multiple clients.
- Designed and implemented an application, which changes colour, shape or picture of caret according the current locale and language. Wrote about 1,000 lines in C.
- Designed and implemented a system-level Windows DLL for synchronizing screen menu, windows and messages with synthesized speech. Wrote about 1,000 lines in C.

Languages: C, C++, Visual Basic, Java, Perl, JavaScript
Tools: Visual C++, Visual Basic, Visual J++, Visual InterDev, MFC, Win32 SDK, Borland C++
Technologies: Imaging for Windows, COM, OLE, Automation, ActiveX, CGI, RDBMS, Client-server, sockets

Independent Contractor (Software Engineer) at Agama (Russia)
October 1994 - October 1995 (1 year 1 month)

Designed and implemented a word processor for Windows 3.x (as a part of a spelling and grammar correction system) and a DLL for spelling in every Windows application. Wrote about 18,000 lines in C.

Languages: C
Tools: Borland C++, Windows SDK

Software Engineer at Covox
February 1992 - March 1994 (2 years 2 months)

Remote work for a USA company Covox, Inc. located in Eugene, Oregon.

- Implemented a vocabulary editing and training DLL as a part of a speech recognition system for Windows 3.x. Wrote about 5,000 lines in C++.

- Designed and implemented an advanced macro recorder/playback program for Windows 3.x with C-style macro language.

Languages: C, C++
Tools: Borland C++, Windows SDK

Independent Contractor (Software Engineer) at Moscow State University
1987 - 1992 (5 years)

Paid work during my Chemistry studies. Includes contracts with Applied Mathematics and Chemistry Laboratory, Inorganic Chemistry Department, and Small Business Centre.

- Designed and implemented an equation solving system with interpreter for Windows 3.x. Wrote about 4,000 lines in C++.
- Designed and implemented a thin layers diffusion analysis system for MS DOS. Wrote about 3,000 lines in C++.
- Designed and implemented an HPGL MS DOS emulator for IBM PagePrinter. Wrote about 1,000 lines in C.
- Designed and implemented a cluster search and analysis system for a mass spectrometer. Wrote about 5,000 lines in C for MS DOS.
- Ported a program (800 Fortran lines) to an assembler equivalent as a part of a system that calculated properties of cosmos fuel for Russian Space Agency. Wrote about 2,500 PDP-11 assembler lines for RSX-11M.

Languages: C, C++, Fortran, x86 assembly language, PDP-11 assembler
Tools: Borland C++, third-party GUI C++ libraries, Microsoft FORTRAN, TASM, Turbo C

Projects

TestWER
2006 to 2011
Members:Dmitry Vostokov

Idea, design, and implementation. Proper software execution artefact collection configuration is one of the most important tasks in software technical support. This diagnostics tool tests post-mortem debugging and Windows Error Reporting configuration for proper process memory dump collection. Supports testing both software and hardware exceptions, x86, x64, and .NET, first- and second-chance exception processing, GUI and command line options.

Editor (Java)
1997 to 1998
Members:Dmitry Vostokov

Designed and implemented an editor with syntax highlighting and word-processing capabilities including media object embedding. Wrote about 10,000 lines in Java 1.0 using Visual J++. Toolbars, status bar, and ruler were all written from scratch.

StressPrinters

2006 to 2009

Members:Dmitry Vostokov, Ramzy Mansour, Gary Barton, Nicholas Vasile, Julian Petrov

Designed and implemented GUI for a set of multithreaded command line tools by using Tool Façade DebugWare pattern.

Many printer driver problems in Terminal Services/Remote Desktop Services environments revolve around poor multi-threaded performance, which in turn can cause print spooler instability. Problematic multi-threaded performance is usually exposed when multiple users connect to a Terminal Server simultaneously using the same printer driver. Symptoms include the failure to autocreate client printers, increased thread count of the printer spooler and/or Citrix Print Manager services, and possibly the unresponsiveness and/or unexpected termination of these services (stop responding).

This tool can be used to simulate multiple sessions auto-creating printers using the same printer driver.

It can also be used to compare the following among various drivers:

• CPU load incurred while creating a printer using a particular driver

• Time required to create a printer successfully using a particular driver

WindowHistory64

2006 to 2013

Members:Dmitry Vostokov

Idea, design, and implementation. WindowHistory64 can record and save all changes to all windows (record changes check box). The following changes are recorded (including time when the change happens):

- Title
- Position and size
- Placement commands
- Attributes: visible, foreground, hung, minimized, maximized
- Style
- Extended style
- Parent and ID
- Class style
- Owner
- Z-Order

In addition WindowHistory64 records and displays:

- Start and end (format) time
- Records the screen size of primary display monitor
- Indicates whether a window is valid at the formatting time (helps to eliminate destroyed windows)
- Window creation and destruction time (accurate only when real-time is enabled)

A new feature introduced in version 4.0, tool tips showing window information. Point the mouse cursor to any window and tool tips show that window handle, title, class, parent window handle, title and class, process and thread ID.

The 64-bit tool also records information about 32-bit process windows. For pure 32-bit systems there is a 32-bit version.

MessageHistory

2006 to 2013

Members:Dmitry Vostokov

Idea, design, and implementation. MessageHistory records window messages and has the following features:

- Similar simple interface as in WindowHistory64
- Both 32-bit and 64-bit versions (MessageHistory.exe and MessageHistory64.exe)
- 64-bit version can log messages from windows created by native 64-bit applications
- You can run both 32-bit and 64-bit versions at the same time to capture messages from all windows in 64-bit system
- Extremely fast logging
- Option for a circular buffer logging (helps to log messages for intermittent issues)
- Can be used in conjunction with WindowHistory 4.0 for 32-bit platforms or WindowHistory64 4.0 for x64 platforms for troubleshooting complex GUI scenarios
- Formatted output is sorted and includes process and thread ID, window class and title
- Additional Spy++-style log for bulk messages
- Shows keyboard and mouse messages in detail: virtual key and character codes, scan code, various flags and mouse coordinates
- Shows activation/deactivation parameters for focus messages
- Shows parameters, position and width/height for “size” and “move” messages

---

Skills & Expertise

Debugging
Device Drivers
Linux

Reverse Engineering
Win32 API
WinDbg
Citrix
Design Patterns
.NET
Windows Internals
Software Development
C++
UML
Visual Studio
x86 Assembly
Kernel Debugging
Distributed Systems
Software Engineering
TCP/IP
C
System Architecture
Kernel Programming
Malware Analysis
C#
X86_64
Agile Methodologies
Technical Support
Objective-C
Technical Writing
Technical Presentations
Management
Crash Dump Analysis
Published Author
Training
Technical Seminars
Professional Services
Architectural Patterns
Publishing
Training Delivery
GDB
Software Diagnostics
Mac OS X
Android Development
Java
Eclipse
Software Forensics
Xcode
IDA Pro

LLDB
ARM

Languages

English	(Full professional proficiency)
Russian	(Native or bilingual proficiency)
French	(Limited working proficiency)
German	(Limited working proficiency)

Publications

Memory Dump Analysis Anthology, Volume 1
OpenTask April 15, 2008
Authors: Dmitry Vostokov

This is a revised, edited, cross-referenced and thematically organized volume of selected DumpAnalysis.org blog posts about crash dump analysis and debugging written in 2006 - 2007 for software engineers developing and maintaining products on Windows platforms, technical support and escalation engineers dealing with complex software issues and general Windows users.

Memory Dump Analysis Anthology, Volume 2
OpenTask October 3, 2008
Authors: Dmitry Vostokov

This is a revised, edited, cross-referenced and thematically organized volume of selected DumpAnalysis.org blog posts about crash dump analysis and debugging written in January - September 2008 for software engineers developing and maintaining products on Windows platforms, quality assurance engineers testing software on Windows platforms and technical support and escalation engineers dealing with complex software issues. The second volume features: - 45 new crash dump analysis patterns - Pattern interaction and case studies - Updated checklist - Fully cross-referenced with Volume 1 - New appendixes

Memory Dump Analysis Anthology, Volume 3
OpenTask December 18, 2009
Authors: Dmitry Vostokov

This is a revised, edited, cross-referenced and thematically organized volume of selected DumpAnalysis.org blog posts about crash dump analysis and debugging written in October 2008 - June 2009 for software engineers developing and maintaining products on Windows platforms, quality assurance engineers testing software on Windows platforms and technical support and escalation engineers dealing with complex software issues. The third volume features: - 15 new crash dump analysis patterns - 29 new pattern interaction case studies - Trace analysis patterns - Updated checklist - Fully cross-referenced with Volume 1 and Volume 2 - New appendixes

Memory Dump Analysis Anthology, Volume 4
OpenTask November 5, 2010
Authors: Dmitry Vostokov

This is a revised, edited, cross-referenced and thematically organized volume of selected DumpAnalysis.org blog posts about crash dump analysis and debugging written in July 2009 - January 2010 for software engineers developing and maintaining products on Windows platforms, quality assurance engineers testing software on Windows platforms, technical support and escalation engineers dealing with complex software issues, and security researchers, malware analysts and reverse engineers. The fourth volume features: - 15 new crash dump analysis patterns - 13 new pattern interaction case studies - 10 new trace analysis patterns - 6 new Debugware patterns and case study - Workaround patterns - Updated checklist - Fully cross-referenced with Volume 1, Volume 2 and Volume 3 - Memory visualization tutorials - Memory space art

Windows Debugging: Practical Foundations

OpenTask February 1, 2009

Authors: Dmitry Vostokov

Written by the founder of DumpAnalysis.org this book is not about bugs or debugging techniques but about background knowledge everyone needs to start experimenting with WinDbg, learn from practical experience and read other advanced debugging books. Solid understanding of fundamentals like pointers is needed to analyze stack traces beyond !analyze -v and lmv WinDbg commands. This is the book to help technical support and escalation engineers and Windows software testers without the knowledge of assembly language to master necessary prerequisites to understand and start debugging and crash dump analysis on Windows platforms. It doesn't require any specific knowledge, fills the gap and lowers the learning curve. The book is also useful for software engineers coming from managed code or Java background, engineers coming from non-Wintel environments, Windows C/C++ software engineers without assembly language background, security researchers and beginners learning Windows software disassembling and reverse engineering techniques. This book can also be used as Intel assembly language and Windows debugging supplement for relevant undergraduate level courses.

x64 Windows Debugging: Practical Foundations

OpenTask August 7, 2009

Authors: Dmitry Vostokov

Written by the founder of DumpAnalysis.org this book is not about bugs or debugging techniques but about background knowledge everyone needs to start experimenting with x64 WinDbg, learn from practical experience and read other advanced debugging books. Solid understanding of fundamentals like pointers is needed to analyze stack traces beyond !analyze -v and lmv WinDbg commands. This is the book to help technical support and escalation engineers and Windows software testers without the knowledge of assembly language to master necessary prerequisites to understand and start debugging and crash dump analysis on x64 Windows platforms. It doesn't require any specific knowledge, fills the gap and lowers the learning curve. The book is also useful for software engineers coming from managed code or Java background, engineers coming from non- Wintel environments, Windows C/C++ software engineers without assembly language background, security researchers and beginners learning x64 Windows software disassembling and reverse engineering techniques. This book can also be used as AMD64 and Intel EM64T assembly language and x64 Windows debugging supplement for relevant undergraduate level courses. For someone, who wants to learn these foundations in the context of 32-bit Windows environments there is a separate x86

book (ISBN: 978-1-906717-10-0). However, this book is completely independent from that earlier book and almost every illustration was recreated to reflect x64 architecture and x64 Windows ILP 32-32-64 model (Integer-Long-Pointer).

WinDbg: A Reference Poster and Learning Cards

OpenTask November 20, 2008

Authors: Dmitry Vostokov

WinDbg is a powerful debugger from Microsoft Debugging Tools for Windows. It has more than 350 commands that can be used in different debugging scenarios. The cover of this book is a poster featuring crash dump analysis checklist and common patterns seen in memory dumps and live debugging sessions. Inside the book you can find ready to cut learning cards with commands and their descriptions coloured according to their use for crash dump or live debugging sessions and user, kernel or complete memory dumps. Tossing cards can create unexpected connections between commands and help to learn them more quickly. Uncut pages can also serve as birds eye view to WinDbg debugging capabilities. More than 350 WinDbg commands including meta-commands and extensions are included.

Pattern-Driven Memory Dump Analysis WinDbg Command Supplement

Debugged! MZ/PE (MagaZine for/from Practicing Engineers) / OpenTask March 16, 2009

Authors: Dmitry Vostokov

On the 26th of February at a Global Engineering Conference organized by Microsoft Global Escalation Services team I was presenting Pattern-Driven Memory Dump Analysis methodology that involves scripts, checklists and patterns and can be summarized as a waterfall-like diagram to the right of this column. We can use various scripts to get textual information from memory dumps, then we can use various checklists to extract specific information and aid the identification of memory dump analysis patterns, common recurrent identifiable problems together with a set of recommendations and possible solutions to apply in a specific context. We have identified more than 100 patterns[1] and many readers of my blog suggested to map them to WinDbg commands or command combinations used to identify them. This article is a first attempt to do such mapping based on commands I used to describe patterns on my blog and in Memory Dump Analysis Anthology volumes. It can also be used as a command reminder and serve as a list of the most used core WinDbg commands.

Modeling CPU Spikes

Debugged! MZ/PE: Modeling Software Defects / OpenTask June 30, 2009

Authors: Dmitry Vostokov, Konstantin Chebotarev

Once we were discussing various ways to save memory dumps during high CPU consumption and both of us independently had an idea about writing such a small tool which allows to model different types of CPU load with different number of threads and different priorities. Because this tool can help people to understand CPU spike issues and play with different scenarios we decided to write an article for this magazine issue.

Modeling Exception Handling

Debugged! MZ/PE: Modeling Software Defects / OpenTask June 30, 2009

Authors: Dmitry Vostokov

Once we got a process memory dump with a stack overflow pattern[5]. There were hundreds of repeated frames involving exception dispatch ...

Memory Dump and Trace Analysis: A Unified Pattern Approach

Debugged! MZ/PE: Software Tracing / OpenTask September 30, 2009

Authors: Dmitry Vostokov

Only memory dump analysis (static troubleshooting and postmortem debugging) or only software trace analysis (dynamic troubleshooting and postmortem debugging) is not always sufficient to resolve customer problems. Sometimes the combination of both methods provides the best result as the following synthetic case study shows.

Colorimetric Tracing: A Visual Approach to Tracking Function Calls

Debugged! MZ/PE: Software Tracing / OpenTask September 30, 2009

Authors: Dmitry Vostokov

Sometimes we need to know whether a function was called or not. Traditional non-invasive approach without setting and triggering debugger breakpoints is to use diagnostic software tracing and record a message when program execution enters a function. The author applied the method of colorimetric computer memory dating[10] to record the function prolog entrance. The idea is to allocate a static buffer for every function we want to trace and fill it with a characteristic RGB patten upon entrance.

What is an Adjoint Thread?

Debugged! MZ/PE: Multithreading / OpenTask March 31, 2010

Authors: Dmitry Vostokov

Let’s recall the definition of a thread from the Dictionary of Debugging[1]: A mapping T: t -> IP, where t is a discreet time arrow and IP is a memory space (Instruction Pointer). The following diagram depicts a thread in an equi-bipartitional virtual memory space comprised from user and kernel spaces ...

Taxonomy of Windows Threads (Part 1)

Debugged! MZ/PE: Multithreading / OpenTask March 31, 2010

Authors: Dmitry Vostokov

What kinds of threads are in a typical Windows system? The usual answer is kernel and user space threads[4]. The former originate in the kernel space and the latter originate in the user space. User thread stack trace can span both spaces with separate stacks (we highlight kernel space stack trace addresses in red, user space stack trace addresses in blue and taxonomic features in bold green) ...

Two ETW/CDF Analysis Tools: A Comparison

Debugged! MZ/PE: Multithreading / OpenTask March 31, 2010

Authors: Dmitry Vostokov

ETW[5] (and Citrix extension, CDF[6]) traces can be very large having millions of messages. Therefore, it is of great importance to have tools that make analysis efficient and effective. In this article I compare two such tools developed by Citrix technical support and published as CDFControl[7] (version 2.5.0.22) and CDFAnalyzer[8] (version 1.0). Each tool has its own strengths and weaknesses. We start with CDFControl. It was primarily developed as an alternative to Microsoft Tracelog[9] and TraceView[10] tools to capture

traces and view them in real-time. Over time the tool has matured into an offline analysis tool as well. Its main strengths compared to CDFAnalyzer are ...

Book Review: Advanced .NET Debugging

Debugged! MZ/PE: Multithreading / OpenTask March 31, 2010

Authors: Dmitry Vostokov

When learning software debugging it is extremely useful to know both high level and low level pictures, from architecture to code implementation details. Deviations of expected software behaviour originate from many sources and not only from defects in your own code. This is especially true for Microsoft .NET platform. Typical .NET application is actually an operation system process. Therefore, .NET debugging in real life situations sometimes involves normal application level Windows debugging where loaded unmanaged components may influence the container process behaviour. Sometimes we even need to look at system level and know a bit of hard core kernel stuff. Standard middle level .NET debugging tools don't allow you to debug from low level perspective. That's why we need a powerful low-level debugger WinDbg.

Memory Dump Analysis Anthology: Color Supplement for Volumes 1-3

OpenTask May 1, 2010

Authors: Dmitry Vostokov

This is a supplemental volume of selected articles with 68 full color illustrations from Memory Dump Analysis Anthology: revised, edited, cross-referenced and thematically organized volumes of selected DumpAnalysis.org blog posts about modern crash dump analysis and debugging written in August 2006 - June 2009 for software engineers developing and maintaining products on Windows platforms, quality assurance engineers testing software on Windows platforms, technical support and escalation engineers dealing with complex software issues, security and defect researchers, computer scientists and philosophers. Unique in its breadth, depth, and scope it offers unprecedented insight into the world of Windows software and draws profound scientific and metaphysical implications.

DLL List Landscape: The Art from Computer Memory Space

OpenTask November 22, 2008

Authors: Dmitry Vostokov

This is a new full color conceptual art book featuring magnificent images from process user space generated by Dump2Picture.

Memory Dump Analysis Anthology, Volume 5

OpenTask April 17, 2011

Authors: Dmitry Vostokov

Five volumes of cross-disciplinary Anthology (dubbed by the author "The Summa Memorianica") lay the foundation of the scientific discipline of Memoretics (study of computer memory snapshots and their evolution in time) that is also called Memory Dump and Software Trace Analysis.

The 5th volume contains revised, edited, cross-referenced, and thematically organized selected DumpAnalysis.org blog posts about crash dump, software trace analysis and debugging written in February 2010 - October 2010 for software engineers developing and maintaining products on Windows platforms,

quality assurance engineers testing software on Windows platforms, technical support and escalation engineers dealing with complex software issues, and security researchers, malware analysts and reverse engineers. The fifth volume features:

- 25 new crash dump analysis patterns
- 11 new pattern interaction case studies (including software tracing)
- 16 new trace analysis patterns
- 7 structural memory patterns
- 4 modeling case studies for memory dump analysis patterns
- Discussion of 3 common analysis mistakes
- Malware analysis case study
- Computer independent architecture of crash analysis report service
- Expanded coverage of software narratology
- Metaphysical and theological implications of memory dump worldview
- More pictures of memory space and physicalist art
- Classification of memory visualization tools
- Memory visualization case studies
- Close reading of the stories of Sherlock Holmes: Dr. Watson’s observational patterns
- Fully cross-referenced with Volume 1, Volume 2, Volume 3, and Volume 4

A Note about Heap (in Windows Debugging Notebook: Essential User Space WinDbg Commands)
OpenTask May 15, 2011
Authors: Dmitry Vostokov

I'm an editor of Troubleshooting and Debugging Notebooks series. Windows Debugging Notebook: Essential User Space WinDbg Commands written by Roberto Alexis Farah is 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.

Memory Dump Analysis Anthology: Color Supplement for Volumes 4-5
OpenTask June 21, 2011
Authors: Dmitry Vostokov

This is a supplemental volume of selected articles with 170 full color illustrations from Memory Dump Analysis Anthology: revised, edited, cross-referenced and thematically organized volumes of selected DumpAnalysis.org blog posts about debugging, modern crash dump and software trace analysis, conceptual physicalist and memory space art, speculative metaphysics of memory dump worldview (memoidealism) written in July 2009 - October 2010 for software engineers developing and maintaining products on Windows platforms, quality assurance engineers testing software on Windows platforms, technical support and escalation engineers dealing with complex software issues, security and defect researchers, reverse engineers and malware analysts, computer security and cyber warfare intelligence professionals, computer scientists, conceptual digital artists and philosophers. Unique in its breadth, depth, and scope it offers

unprecedented insight into the world of software behavior and draws profound engineering, scientific, artistic and metaphysical implications.

Introduction to Pattern-Driven Software Problem Solving

OpenTask June 21, 2011

Authors: Dmitry Vostokov, Memory Dump Analysis Services

The full transcript of Memory Dump Analysis Services Webinar on pattern-driven software troubleshooting, debugging and maintenance. Topics include: A Short History of DumpAnalysis.org; Memory Dump Analysis Patterns; Troubleshooting and Debugging Tools (Debugware) Patterns; Software Trace Analysis Patterns; From Software Defects to Software Behavior; Workaround Patterns; Structural Memory Patterns; Memory Analysis Domain Pattern Hierarchy; New Directions.

Accelerated Windows Memory Dump Analysis: Training Course Transcript and WinDbg Practice Exercises with Notes

OpenTask August 7, 2011

Authors: Dmitry Vostokov, Memory Dump Analysis Services

The full transcript of Memory Dump Analysis Services Training with 21 step-by-step exercises, notes, source code of specially created modeling applications and selected Q&A. Covers about 50 crash dump analysis patterns from process, kernel and complete memory dumps.

Accelerated .NET Memory Dump Analysis: Training Course Transcript and WinDbg Practice Exercises with Notes

OpenTask November 14, 2011

Authors: Dmitry Vostokov, Memory Dump Analysis Services

The full transcript of Memory Dump Analysis Services Training with 7 step-by-step exercises, notes, source code of specially created modeling applications and selected Q&A. Covers 20 .NET memory dump analysis patterns plus additional unmanaged patterns.

Advanced Windows Memory Dump Analysis with Data Structures: Training Course Transcript and WinDbg Practice Exercises with Notes

OpenTask February 20, 2012

Authors: Dmitry Vostokov, Memory Dump Analysis Services

The full transcript of Memory Dump Analysis Services Training with 10 step-by-step exercises, notes, and selected Q&A.

Software Narratology: An Introduction to the Applied Science of Software Stories

OpenTask April 15, 2012

Authors: Dmitry Vostokov, Memory Dump Analysis Services

This is a transcript of Memory Dump Analysis Services Webinar about Software Narratology: an exciting new discipline and a field of research founded by DumpAnalysis.org. When software executes it gives us its stories in the form of UI events, software traces and logs. Such stories can be analyzed for their structure and patterns for troubleshooting, debugging and problem resolution purposes. Topics also include software narremes and their types, anticipatory software construction and software diagnostics.

Pattern-Driven Memory Dump Analysis - WinDbg Command Supplement

Debugged! MZ/PE magazine, March, 2009 issue/OpenTask March 16, 2009

Authors: Dmitry Vostokov

The article maps memory dump analysis patterns to WinDbg commands or command combinations used to identify them.

Accelerated Mac OS X Core Dump Analysis: Training Course Transcript and GDB Practice Exercises

OpenTask August 15, 2012

Authors: Dmitry Vostokov, Memory Dump Analysis Services, Software Diagnostics Services

The full transcript of Memory Dump Analysis Services training with 12 step-by-step exercises. Learn how to analyze app crashes and freezes, navigate through process core memory dump space and diagnose corruption, memory leaks, CPU spikes, blocked threads, deadlocks, wait chains, and much more. We use a unique and innovative pattern-driven analysis approach to speed up the learning curve. The training consists of practical step-by-step exercises using Xcode and GDB environments highlighting more than 30 patterns diagnosed in 64-bit process core memory dumps. The training also includes an overview of relevant similarities and differences between Windows and Mac OS X user space memory dump analysis useful for engineers with Wintel background. Audience: Software technical support and escalation engineers, system administrators, software developers, security professionals and quality assurance engineers.

Accelerated Windows Software Trace Analysis: Training Course Transcript

OpenTask December 27, 2012

Authors: Dmitry Vostokov, Software Diagnostics Services

The full transcript of Software Diagnostics Services software trace and log analysis training.

Memory Dump Analysis Anthology, Volume 6

OpenTask January 25, 2013

Authors: Dmitry Vostokov

Contains revised, edited, cross-referenced, and thematically organized selected DumpAnalysis.org blog posts about memory dump and software trace analysis, software troubleshooting and debugging written in November 2010 - October 2011 for software engineers developing and maintaining products on Windows platforms, quality assurance engineers testing software on Windows platforms, technical support and escalation engineers dealing with complex software issues, and security researchers, malware analysts and reverse engineers.

Accelerated Windows Malware Analysis with Memory Dumps: Training Course Transcript and WinDbg Practice Exercises

OpenTask February 16, 2013

Authors: Dmitry Vostokov, Software Diagnostics Services

The full transcript of Software Diagnostics Services training with exercises covering more than 20 malware analysis patterns.

Accelerated Windows Memory Dump Analysis: Training Course Transcript and WinDbg Practice Exercises with Notes, Second Edition

OpenTask March 1, 2013

Authors: Dmitry Vostokov, Software Diagnostics Services

The full transcript of Software Diagnostics Services training with 23 step-by-step exercises, notes, source code of specially created modeling applications and more than 100 questions and answers. Covers more than 50 crash dump analysis patterns diagnosed in 32-bit and 64-bit process, kernel and complete memory dumps. Learn how to analyze application, service and system crashes and freezes, navigate through memory dump space and diagnose heap corruption, memory leaks, CPU spikes, blocked threads, deadlocks, wait chains, and much more. The training uses a unique and innovative pattern-driven analysis approach to speed up the learning curve. Prerequisites: Basic Windows troubleshooting. Audience: software technical support and escalation engineers, system administrators, security professionals, software developers and quality assurance engineers.

Pattern-Driven Software Diagnostics: An Introduction

OpenTask April 2, 2013

Authors: Dmitry Vostokov, Software Diagnostics Services

This is a transcript of Software Diagnostics Services Webinar about different pattern categories for effective and efficient abnormal software behavior diagnostics: the foundation of scalable and cost-effective pattern-driven software support.

Accelerated Windows Debugging 3: Training Course Transcript and WinDbg Practice Exercises

OpenTask July 27, 2013

Authors: Dmitry Vostokov, Software Diagnostics Services

The full transcript of Software Diagnostics Services live debugging training.

Systemic Software Diagnostics: An Introduction

OpenTask July 30, 2013

Authors: Dmitry Vostokov, Software Diagnostics Services

This is a transcript of Software Diagnostics Services Webinar about how to apply systems theory and systems thinking for effective and efficient abnormal software behaviour diagnostics: the foundation of software troubleshooting and debugging.

Accelerated .NET Memory Dump Analysis: Training Course Transcript and WinDbg Practice Exercises, Second Edition

OpenTask August 10, 2013

Authors: Dmitry Vostokov, Software Diagnostics Services

The full transcript of Software Diagnostics Services training with 9 step-by-step exercises, notes, source code of specially created modeling applications and selected Q&A. Covers 20 .NET memory dump analysis patterns plus additional unmanaged patterns.

Software Diagnostics: The Collected Seminars

OpenTask September 11, 2013

Authors: Dmitry Vostokov, Software Diagnostics Services

This is a collection of Software Diagnostics Services webinar transcripts about pattern-oriented software diagnostics developed by Software Diagnostics Institute. Includes 9 seminars on pattern-driven software problem solving, software narratology, pattern-driven software diagnostics, systemic software diagnostics,

pattern-based software diagnostics, philosophy of software diagnostics, victimware, malware narratives and pattern-oriented network trace analysis.

Pattern-Based Software Diagnostics: An Introduction

OpenTask August 23, 2013

Authors: Dmitry Vostokov, Software Diagnostics Services

This is a transcript of Software Diagnostics Services Webinar about how pattern-based software diagnostics, troubleshooting and debugging address software post-construction problem solving pattern life cycle: from the discovery of a new pattern through its integration into an existing pattern catalogue and pattern language, testing, packaging and delivering to pattern consumers with subsequent usage, refactoring and writing case studies.

Victimware: The Missing Part of the Equation

OpenTask August 27, 2013

Authors: Dmitry Vostokov, Software Diagnostics Services

Some software components are innocent victims of other component coding mistakes or deliberate subversion and some start as a part of crimeware and malware but eventually become victims themselves (they crash, hang, spike, leak, are dumped, subverted, etc.) This is a transcript of Software Diagnostics Services Webinar about unified malware and victimware analysis by using behavioural and structural patterns including a live memory dump analysis example.

Philosophy of Software Diagnostics: An Introduction, Part 1

OpenTask September 1, 2013

Authors: Dmitry Vostokov, Software Diagnostics Services

This is a transcript of Software Diagnostics Services Webinar about phenomenological, hermeneutical and analytical approaches to software diagnostics.

Malware Narratives: An Introduction

OpenTask September 5, 2013

Authors: Dmitry Vostokov, Software Diagnostics Services

Software Narratology, the science of software stories, found its successful application in software diagnostics of abnormal software behaviour, especially in the pattern-driven and pattern-based analysis of software logs from complex systems with millions of events, thousands of threads, hundreds of processes and modules. This is a transcript of Software Diagnostics Services Webinar on the new application of software narratology to malware analysis.

Pattern-Oriented Network Trace Analysis

OpenTask September 7, 2013

Authors: Dmitry Vostokov, Software Diagnostics Services

Software Narratology found its successful application in software diagnostics of abnormal software behaviour in software logs. This is a transcript of Software Diagnostics Services Webinar on the new application of software narratology to network trace analysis with examples from Wireshark.

Mobile Software Diagnostics: An Introduction

OpenTask September 19, 2013

Authors: Dmitry Vostokov, Software Diagnostics Services

This is a transcript of Software Diagnostics Services Webinar about the perspectives of pattern-oriented software diagnostics in mobile world with examples for Android and Java.

The Old New Crash: Cloud Memory Dump Analysis

OpenTask August 1, 2011

Authors: Dmitry Vostokov, Software Diagnostics Services

This is a transcript of Software Diagnostics Services (former Memory Dump Analysis Services) Webinar about a uniform methodology and tools for analysis of crashes, hangs, and other types of abnormal software behaviour in cloud environments.

Accelerated Disassembly, Reconstruction and Reversing: Training Course Transcript and WinDbg Practice Exercises with Memory Cell Diagrams

OpenTask November 5, 2013

Authors: Dmitry Vostokov, Software Diagnostics Services

The full transcript of Software Diagnostics Services training with 6 step-by-step exercises, notes, source code of specially created modeling applications, memory cell diagrams and selected Q&A. Covers mote than 25 ADDR patterns.

Accelerated Windows Memory Dump Analysis: Training Course Transcript and WinDbg Practice Exercises with Notes, Third Edition

OpenTask November 25, 2013

Authors: Dmitry Vostokov, Software Diagnostics Services

The full transcript of Software Diagnostics Services training with 25 step-by-step exercises, notes, source code of specially created modelling applications and more than 100 questions and answers. Covers more than 50 crash dump analysis patterns diagnosed in 32-bit and 64-bit process, kernel and complete memory dumps. Learn how to analyse application, service and system crashes and freezes, navigate through memory dump space and diagnose heap corruption, memory leaks, CPU spikes, blocked threads, deadlocks, wait chains, and much more. The training uses a unique and innovative pattern-driven analysis approach to speed up the learning curve. Prerequisites: Basic Windows troubleshooting. Audience: software technical support and escalation engineers, system administrators, security professionals, software developers and quality assurance engineers. The 3rd edition was updated to the latest version of WinDbg from Debugging Tools for Windows and includes news exercises for Windows 7 and Windows 8.1 crash dumps.

Advanced Windows Memory Dump Analysis with Data Structures: Training Course Transcript and WinDbg Practice Exercises with Notes, Second Edition

OpenTask December 15, 2013

Authors: Dmitry Vostokov, Software Diagnostics Services

The full transcript of Memory Dump Analysis Services Training with 10 step-by-step exercises, notes, and selected questions and answers. Learn how to navigate through memory dump space and Windows data structures to troubleshoot and debug complex software incidents. The training uses a unique and innovative pattern-driven analysis approach to speed up the learning curve. It consists of practical step-by-step exercises using WinDbg to diagnose structural and behavioural patterns in 64-bit kernel and complete

(physical) memory dumps. Additional topics include memory search, kernel linked list navigation, practical WinDbg scripting, registry, system variables and objects, device drivers and I/O. Prerequisites are basic and intermediate level Windows memory dump analysis: ability to list processors, processes, threads, modules, apply symbols, walk through stack traces and raw stack data, diagnose patterns such as heap corruption, CPU spike, memory and handle leaks, access violation, stack overflow, critical section and resource wait chains and deadlocks. If you don't feel comfortable with prerequisites then Accelerated Windows Memory Dump Analysis training book is recommended before purchasing and reading this book course. Audience: Software developers, software technical support and escalation engineers, reverse and security research engineers. The 2nd edition contains updated exercises for the latest WinDbg version from Windows SDK 8.1.

Pattern-Oriented Software Forensics: A Foundation of Memory Forensics and Forensics of Things

OpenTask February 4, 2014

Authors: Dmitry Vostokov, Software Diagnostics Services

This is a transcript of Software Diagnostics Services Webinar about a comprehensive theory behind software forensics based on systemic and pattern-oriented software diagnostics developed by Software Diagnostics Institute. It synthesises pattern-oriented memory analysis of malware and victimware with pattern-oriented software log and trace analysis based on software narratology.

Fundamentals of Physical Memory Analysis

OpenTask February 6, 2014

Authors: Dmitry Vostokov, Software Diagnostics Services

This is a transcript of Software Diagnostics Services Webinar about physical memory analysis on desktop and server Windows platforms (a revised version of the previous webinar on complete crash and hang memory dump analysis). Topics include: memory acquisition and its tricks; user vs. kernel vs. physical memory space; fibre bundle space; challenges of physical memory analysis; common WinDbg commands; patterns; common mistakes; a hands-on analysis example with logs; a guide to further study.

Accelerated Mac OS X Core Dump Analysis, Second Edition: Training Course Transcript with GDB and LLDB Practice Exercises

OpenTask March 1, 2014

Authors: Dmitry Vostokov, Software Diagnostics Services

The full transcript of Software Diagnostics Services (former Memory Dump Analysis Services) training with 12 step-by-step exercises. Learn how to analyse app crashes and freezes, navigate through process core memory dump space and diagnose corruption, memory leaks, CPU spikes, blocked threads, deadlocks, wait chains, and much more. We use a unique and innovative pattern-driven analysis approach to speed up the learning curve. The training consists of practical step-by-step exercises using GDB and LLDB debuggers highlighting more than 30 memory analysis patterns diagnosed in 64-bit process core memory dumps. The training also includes source code of modelling applications written in Xcode environment, a catalogue of relevant patterns from Software Diagnostics Institute, and an overview of relevant similarities and differences between Windows and Mac OS X user space memory dump analysis useful for engineers with Wintel background. Audience: Software technical support and escalation engineers, system administrators, software developers, security professionals and quality assurance engineers.

Accelerated Mac OS X Core Dump Analysis: LLDB Exercises

OpenTask March 3, 2014

Authors: Dmitry Vostokov, Software Diagnostics Institute

This is an update for Accelerated Mac OS X Core Dump Analysis: Training Course Transcript and GDB Practice Exercises (ISBN: 978-1908043405) book. In Mac OS X Mavericks GDB was replaced by LLDB debugger. All GDB exercises were reworked and updated for LLDB. The original first edition also contains slide transcripts and selected memory analysis pattern descriptions which are missing in this update. This update contains only LLDB exercises. If you don't have the first edition of this course then Accelerated Mac OS X Core Dump Analysis, Second Edition: Training Course Transcript with GDB and LLDB Practice Exercises (ISBN: 978-1908043719) is recommended instead of this update.

Advanced Windows RT Memory Dump Analysis, ARM Edition: Training Course Transcript and WinDbg Practice Exercises

OpenTask March 18, 2014

Authors: Dmitry Vostokov, Software Diagnostics Services

The full transcript of Software Diagnostics Services training with 9 step-by-step exercises. Learn how to navigate through memory dump space and Windows data structures to perform memory forensics, troubleshoot and debug complex software incidents. The training uses a unique and innovative pattern-driven analysis approach to speed up the learning curve. It consists of practical step-by-step exercises using WinDbg to diagnose structural and behavioural patterns in Windows RT kernel and complete (physical) memory dumps. Additional topics include memory search, kernel linked list navigation, practical WinDbg scripting, registry, system variables and objects, device drivers and I/O, memory mapped and cached files content. Prerequisites are basic and intermediate level Windows memory dump analysis: ability to list processors, processes, threads, modules, apply symbols, and walk through stack traces. Audience: Software developers, software technical support and escalation engineers, reverse and security research engineers, digital forensic analysts.

Memory Dump Analysis Anthology, Volume 7

OpenTask June 8, 2014

Authors: Dmitry Vostokov, Software Diagnostics Institute

Contains revised, edited, cross-referenced, and thematically organized selected articles from Software Diagnostics Institute (DumpAnalysis.org + TraceAnalysis.org) and Software Diagnostics Library (former Crash Dump Analysis blog, DumpAnalysis.org/blog) about software diagnostics, debugging, crash dump analysis, software trace and log analysis, malware analysis and memory forensics written in November 2011 - May 2014 for software engineers developing and maintaining products on Windows (WinDbg) and Mac OS X (GDB) platforms, quality assurance engineers testing software, technical support and escalation engineers dealing with complex software issues, security researchers, malware analysts, reverse engineers, and memory forensics analysts. The seventh volume features:

- 66 new crash dump analysis patterns
- 46 new software log and trace analysis patterns

- 18 core memory dump analysis patterns for Mac OS X and GDB
- 10 malware analysis patterns
- Additional unified debugging pattern
- Additional user interface problem analysis pattern
- Additional pattern classification including memory and log acquisition patterns
- Additional .NET memory analysis patterns
- Introduction to software problem description patterns
- Introduction to software diagnostics patterns
- Introduction to general abnormal structure and behaviour patterns
- Introduction to software disruption patterns
- Introduction to static code analysis patterns
- Introduction to network trace analysis patterns
- Introduction to software diagnostics report schemes
- Introduction to elementary software diagnostics patterns
- Introduction to patterns of software diagnostics architecture
- Introduction to patterns of disassembly, reconstruction and reversing
- Introduction to vulnerability analysis patterns
- Fully cross-referenced with Volume 1, Volume 2, Volume 3, Volume 4, Volume 5, and Volume 6

Memory Dump Analysis Anthology: Color Supplement for Volumes 6-7

OpenTask June 15, 2014

Authors: Dmitry Vostokov, Software Diagnostics Institute

This is a supplemental volume of 150 full color illustrations from Memory Dump Analysis Anthology: revised, edited, cross-referenced, and thematically organized selected articles from Software Diagnostics Institute (DumpAnalysis.org + TraceAnalysis.org) and Software Diagnostics Library (former Crash Dump Analysis blog, DumpAnalysis.org/blog) about software diagnostics, debugging, crash dump analysis, software narratology, software trace and log analysis, malware analysis and memory forensics written in November 2011 - May 2014 for software engineers developing and maintaining software products, quality assurance engineers testing software, technical support and escalation engineers dealing with complex software issues, security researchers, malware analysts, reverse engineers, digital forensics analysts, computer security and cyber warfare intelligence professionals, computer scientists, conceptual digital artists, and philosophers. Unique in its breadth, depth, and scope it offers unprecedented insight into the world of software behavior and draws profound engineering, scientific, artistic, and philosophical implications.

Pattern-Oriented Memory Forensics: A Pattern Language Approach

OpenTask September 23, 2014

Authors: Dmitry Vostokov, Software Diagnostics Institute, Software Diagnostics Services

This is a full-colour transcript of a lecture which introduces a pattern language for memory forensics - investigation of past software behaviour in memory snapshots. It provides a unified language for discussing and communicating detection and analysis results despite the proliferation of operating systems and tools, a base language for checklists, and an aid in accelerated learning. The lecture has a short theoretical part and

then illustrates various patterns seen in crash dumps by using WinDbg debugger from Microsoft Debugging Tools for Windows.

Software Trace and Memory Dump Analysis: Patterns, Tools, Processes and Best Practices

OpenTask September 27, 2014

Authors: Dmitry Vostokov, Software Diagnostics Services

This is a full-colour transcript of Software Diagnostics Services (former Memory Dump Analysis Services) lecture delivered at E2EVC Virtualization Conference in 2011 which introduces an analysis methodology for software execution artefacts.

Principles of Memory Dump Analysis: The Collected Seminars

OpenTask September 28, 2014

Authors: Dmitry Vostokov, Software Diagnostics Services, Software Diagnostics Institute

This is a collection of Software Diagnostics Services webinar transcripts about memory dump analysis methodology developed by Software Diagnostics Institute. Includes 6 seminars on physical memory dump analysis, cloud memory dump analysis, patterns, tools, processes and best practices for software trace and memory dump analysis, pattern-oriented software forensics, a pattern language for memory forensics, and mobile software diagnostics.

Memory Dump Analysis Anthology, Volume 8a

OpenTask November 30, 2014

Authors: Dmitry Vostokov, Software Diagnostics Institute

Contains revised, edited, cross-referenced, and thematically organized selected articles from Software Diagnostics Institute (DumpAnalysis.org + TraceAnalysis.org) and Software Diagnostics Library (former Crash Dump Analysis blog, DumpAnalysis.org/blog) about software diagnostics, debugging, crash dump analysis, memory forensics, software trace and log analysis written in June 2014 - November 2014 for software engineers developing and maintaining products on Windows platforms, quality assurance engineers testing software, technical support and escalation engineers dealing with complex software issues, security researchers, reverse engineers, malware and memory forensics analysts. This volume is fully cross-referenced with volumes 1 - 7 and features:

- 19 new crash dump analysis patterns
- 10 new software log and trace analysis patterns
- Introduction to malnarratives and higher-order pattern narratives
- Introduction to pattern language for performance analysis
- Introduction to pater-oriented debugging process

Pattern-Oriented Software Diagnostics, Debugging, Malware Analysis, Reversing: Sample Training Exercises

Software Diagnostics Services February 3, 2015

Authors: Dmitry Vostokov, Software Diagnostics Services

Free eBook with 9 sample exercises from various training courses developed by Software Diagnostics Services and published by OpenTask.

Software Trace and Log Analysis: A Pattern Reference

OpenTask February 9, 2015

Authors: Dmitry Vostokov, Software Diagnostics Institute

General trace and log analysis patterns allow application of uniform problem detection and solving approach across diverse software environments. This pattern language covers any execution artifact from a small debugging trace to a distributed log with billions of messages from hundreds of computers, thousands of software components, threads, and processes. Pattern-oriented trace and log analysis is applicable to troubleshooting and debugging Windows, Mac OS X, Linux, FreeBSD, Android, iOS, z/OS, and any other possible computer platform. Its pattern catalog is a part of pattern-oriented software diagnostics, forensics, and prognostics developed by Software Diagnostics Institute (DumpAnalysis.org + TraceAnalysis.org). This reference reprints with corrections 100 patterns originally published in Memory Dump Analysis Anthology volumes 3 - 8a and Software Diagnostics Library (former Crash Dump Analysis blog, DumpAnalysis.org/blog). Full-color diagrams accompany most pattern descriptions.

Encyclopedia of Crash Dump Analysis Patterns: Detecting Abnormal Software Structure and Behavior in Computer Memory

OpenTask March 9, 2015

Authors: Dmitry Vostokov, Software Diagnostics Institute

This reference reprints with corrections, additional comments, and classification 326 alphabetically arranged and cross-referenced memory analysis patterns originally published in Memory Dump Analysis Anthology volumes 1 – 8. This pattern catalog is a part of pattern-oriented software diagnostics, forensics, and prognostics developed by Software Diagnostics Institute (DumpAnalysis.org + TraceAnalysis.org). Most of the patterns are illustrated with examples for WinDbg from Debugging Tools for Windows with a few examples from Mac OS X for GDB.

Resume and CV: As a Book

OpenTask November 17, 2008

Authors: Dmitry Vostokov

A book can serve the role of CV, but a CV can serve the role of a book. This is an old CV (1987 - 2003) from the founder of Software Diagnostics Institute (DumpAnalysis.org + TraceAnalysis.org) as an example of a person with a CV-writing obsession.

Memory Dump Analysis Anthology, Volume 8b

OpenTask July 14, 2015

Authors: Dmitry Vostokov, Software Diagnostics Institute

Contains revised, edited, cross-referenced, and thematically organized selected articles from Software Diagnostics Institute (DumpAnalysis.org + TraceAnalysis.org) and Software Diagnostics Library (former Crash Dump Analysis blog, DumpAnalysis.org/blog) about software diagnostics, debugging, crash dump analysis, memory forensics, software trace and log analysis written in December 2014 - July 2015 for software engineers developing and maintaining products on Windows platforms, quality assurance engineers testing software, technical support and escalation engineers dealing with complex software issues, security

researchers, reverse engineers, malware and memory forensics analysts. This volume is fully cross-referenced with volumes 1 – 7, 8a, and features:

- 12 new crash dump analysis patterns
- 15 new software log and trace analysis patterns
- New memory dump analysis case study
- Introduction to articoding
- Introduction to special and general trace and log analysis
- Introduction to projective debugging
- Introduction to artifact-malware
- Introduction to concrete and general problem analysis patterns

Practical Foundations of Windows Debugging, Disassembling, Reversing: Training Course

OpenTask October 10, 2015

Authors: Dmitry Vostokov, Software Diagnostics Services

This training course is a combined and reformatted version of the two previous books Windows Debugging: Practical Foundations and x64 Windows Debugging: Practical Foundations. The new format makes it easy to switch between and compare x86 and x64 versions. The book also has a larger format similar to other training courses from Software Diagnostics Services, punctuation and code highlighting improvements, the output and screenshots from the latest WinDbg 10, and consistently uses WinDbg (X86) for 32-bit examples and WinDbg (X64) for 64-bit examples.

Accelerated Linux Core Dump Analysis: Training Course Transcript with GDB Practice Exercises

OpenTask December 19, 2015

Authors: Dmitry Vostokov, Software Diagnostics Service

The full transcript of Software Diagnostics Services training with 13 step-by-step exercises. Learn how to analyse Linux process crashes and hangs, navigate through process core memory dump space and diagnose corruption, memory leaks, CPU spikes, blocked threads, deadlocks, wait chains, and much more. We use a unique and innovative pattern-oriented diagnostic analysis approach to speed up the learning curve. The training consists of practical step-by-step exercises using GDB debugger highlighting more than 25 memory analysis patterns diagnosed in 64-bit process core memory dumps. The training also includes source code of modelling applications, a catalogue of relevant patterns from Software Diagnostics Institute, and an overview of relevant similarities and differences between Windows and Linux user space memory dump analysis useful for engineers with Wintel background.

Memory Dump Analysis Anthology, Volume 9a

OpenTask February 20, 2016

Authors: Dmitry Vostokov, Software Diagnostics Institute

Contains revised, edited, cross-referenced, and thematically organized selected articles from Software Diagnostics Institute (DumpAnalysis.org + TraceAnalysis.org) and Software Diagnostics Library (former Crash Dump Analysis blog, DumpAnalysis.org/blog) about software diagnostics, root cause analysis, debugging, crash and hang dump analysis, software trace and log analysis written in August 2015 - February

2016 for software engineers developing and maintaining products on Windows platforms, quality assurance engineers testing software, technical support and escalation engineers dealing with complex software issues, security researchers, reverse engineers, malware and memory forensics analysts. This volume is fully cross-referenced with volumes 1 – 8 and features:

- 9 new crash dump analysis patterns
- 9 new software log and trace analysis patterns
- 15 Linux core dump analysis pattern variants
- New workaround pattern
- New memory dump analysis case study
- Introduction to pattern-oriented software internals, pattern paradigms, pattern stacks, pattern repertoire
- Introduction to software diagnostics canvas
- Introduction to patterns-based root cause analysis methodology
- Introduction to a protein metaphor for software traces and logs
- Definition of software diagnostics scope
- Introduction to artificial debugger and pseudo-memory dumps
- Definition of tool-centric and pattern-centric software diagnostics, forensics, prognostics

Accelerated Windows Memory Dump Analysis: Training Course Transcript and WinDbg Practice Exercises with Notes, Fourth Edition

OpenTask May 22, 2016

Authors: Dmitry Vostokov, Software Diagnostics Services

The full transcript of Software Diagnostics Services training with 28 step-by-step exercises, notes, source code of specially created modeling applications and more than 100 questions and answers. Covers more than 60 crash dump analysis patterns from x86 and x64 process, kernel, complete (physical), and active memory dumps. The 4th edition was fully reworked to use WinDbg 10 and now covers memory dumps from Windows 10 x64. It also includes optional legacy exercises from the previous editions covering Windows Vista and Windows 7.

Memory Dump Analysis Anthology, Volume 9b

OpenTask October 2, 2016

Authors: Dmitry Vostokov, Software Diagnostics Institute

Contains revised, edited, cross-referenced, and thematically organized selected articles from Software Diagnostics Institute (DumpAnalysis.org + TraceAnalysis.org) and Software Diagnostics Library (former Crash Dump Analysis blog, DumpAnalysis.org/blog) about software diagnostics, root cause analysis, debugging, crash and hang dump analysis, software trace and log analysis written in March - September 2016 for software engineers developing and maintaining products on Windows platforms, quality assurance engineers testing software, technical support and escalation engineers dealing with complex software issues, security researchers, reverse engineers, malware and memory forensics analysts. This volume is fully cross-referenced with volumes 1 – 9a and features:

- 11 new crash dump analysis patterns
- 11 new software log and trace analysis patterns
- New structural memory pattern
- Introduction to Riemann root cause analysis language
- Introduction to problem solving as code
- Introduction to Dia|gram graphical diagnostic analysis language
- Introduction to iterative pattern-oriented root cause analysis
- Definition of theoretical software diagnostics

Theoretical Software Diagnostics: Collected Articles

OpenTask October 28, 2016

Authors: Dmitry Vostokov, Software Diagnostics Institute

Contains reprinted articles in full color from Memory Dump Analysis Anthology related to pattern-oriented software diagnostics with additional comments showing the historical development of this autonomous and distinctive discipline over the last 10 years.

Software Trace and Log Analysis: A Pattern Reference, Second Edition

OpenTask November 12, 2016

Authors: Dmitry Vostokov, Software Diagnostics Institute

General trace and log analysis patterns allow application of uniform problem detection and solving approach across diverse software environments. This pattern language covers any execution artifact from a small debugging trace to a distributed log with billions of messages from hundreds of computers, thousands of software components, threads, and processes. Pattern-oriented trace and log analysis is applicable to troubleshooting and debugging Windows, Mac OS X, Linux, FreeBSD, Android, iOS, z/OS, and any other possible computer platform. Its pattern catalog is a part of pattern-oriented software diagnostics, forensics, prognostics, root cause analysis, and debugging developed by Software Diagnostics Institute (DumpAnalysis.org + TraceAnalysis.org). This reference reprints with corrections 133 patterns originally published in Memory Dump Analysis Anthology volumes 3 - 9 and Software Diagnostics Library (former Crash Dump Analysis blog, DumpAnalysis.org/blog). Full-color diagrams accompany most pattern descriptions. The second edition includes 33 more patterns and improved pattern index.

Encyclopedia of Crash Dump Analysis Patterns: Detecting Abnormal Software Structure and Behavior in Computer Memory, Second Edition

OpenTask March 11, 2017

Authors: Dmitry Vostokov

This reference reprints with corrections, additional comments, and classification 373 alphabetically arranged and cross-referenced memory analysis patterns originally published in Memory Dump Analysis Anthology volumes 1 – 9 including 5 analysis patterns from volume 10a. This pattern catalog is a part of pattern-oriented software diagnostics, forensics, prognostics, root cause analysis, and debugging developed by Software Diagnostics Institute (DumpAnalysis.org + TraceAnalysis.org). Most of the analysis patterns are illustrated with examples for WinDbg from Debugging Tools for Windows with a few examples from Mac

OS X and Linux for GDB. The second edition includes more than 50 new analysis patterns and more than 70 new examples and comments for analysis patterns published in the first edition.

Certifications

Windows Internals
Microsoft July 2008

OMG Certified UML Professional
Object Management Group August 2003

Education

Lomonosov Moscow State University (MSU)
Chemistry, 1987 - 1992

Kalashnikov Izhevsk State Technical University (ISTU)
BSc/MSc, Computer Science, Software Engineering, 2003 - 2007

Interests

Reading: History, Sociology, Philosophy, Mathematics, Literary Theory and Criticism, Narratology, Physics, Chemistry, Biology, Medicine

Dmitry Vostokov

Pattern-Oriented Software Diagnostics, Forensics, Prognostics, Root Cause Analysis, Debugging. Diagnostics of Things.

dmitry.vostokov@patterndiagnostics.com

---

1 person has recommended Dmitry

> "Dmitry's books and blogs have been instrumental in the analysis of crash dump files. The use of patterns and pattern recognition is a unique, viable approach."
>
> **— Herbert Marshall Rody**, was with another company when working with Dmitry at Software Diagnostics Institute

Contact Dmitry on LinkedIn