Defect

Detect

Windows Malware Analysis Accelerated

with Memory Dumps

Version 3.0

Dmitry Vostokov

Software Diagnostics Services

Published by OpenTask, Republic of Ireland

OpenTask books and magazines are available through booksellers and distributors worldwide. For further information or comments, send requests to press@opentask.com.

A CIP catalog record for this book is available from the British Library.

ISBN-l3: 978-1-912636-96-9 (Paperback)

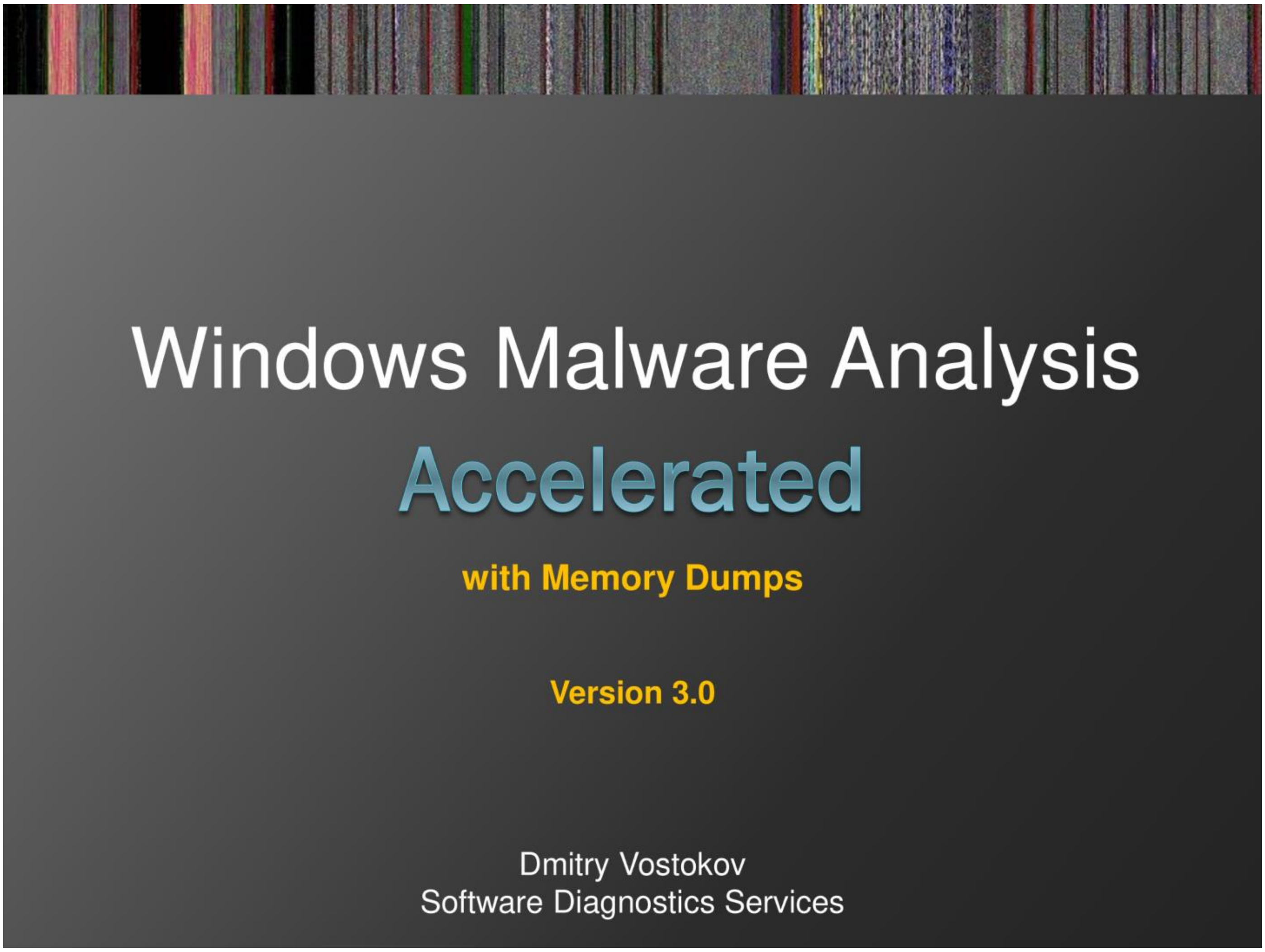
Revision 3.02 (October 2022)

Contents

About the Author

Dmitry Vostokov is an internationally recognized expert, speaker, educator, scientist, inventor, and author. He is the founder of the pattern-oriented software diagnostics, forensics, and prognostics discipline (Systematic Software Diagnostics), and Software Diagnostics Institute (DA+TA: DumpAnalysis.org + TraceAnalysis.org). Vostokov has also authored more than 50 books on software diagnostics, anomaly detection and analysis, software and memory forensics, root cause analysis and problem solving, memory dump analysis, debugging, software trace and log analysis, reverse engineering, and malware analysis. He has over 25 years of experience in software architecture, design, development, and maintenance in various industries, including leadership, technical, and people management roles. Dmitry also founded Syndromatix, Anolog.io, BriteTrace, DiaThings, Logtellect, OpenTask Iterative and Incremental Publishing (OpenTask.com), Software Diagnostics Technology and 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, its further development as Narratology of Things and Diagnostics of Things (DoT), Software Pathology, and Quantum Software Diagnostics. His current interest areas are theoretical software diagnostics and its mathematical and computer science foundations, application of formal logic, artificial intelligence, machine learning and data mining to diagnostics and anomaly detection, software diagnostics engineering and diagnostics-driven development, diagnostics workflow and interaction. Recent interest areas also include cloud native computing, security, automation, functional programming, and applications of category theory to software development and big data.

Introduction



Windows Malware Analysis

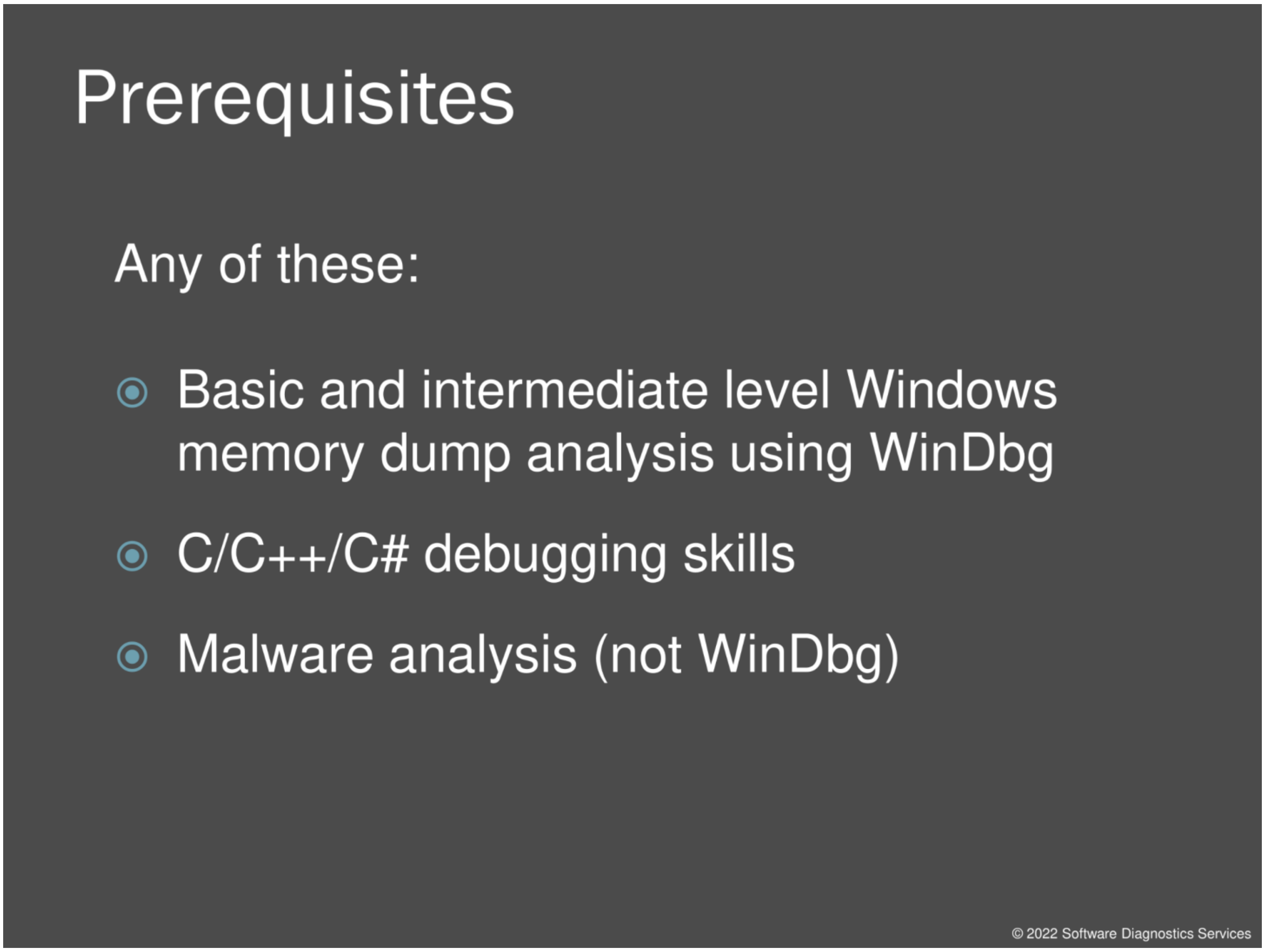
Accelerated

with Memory Dumps

Version 3.0

Dmitry Vostokov
Software Diagnostics Services

Hello everyone, my name is Dmitry Vostokov, and I teach this training course.

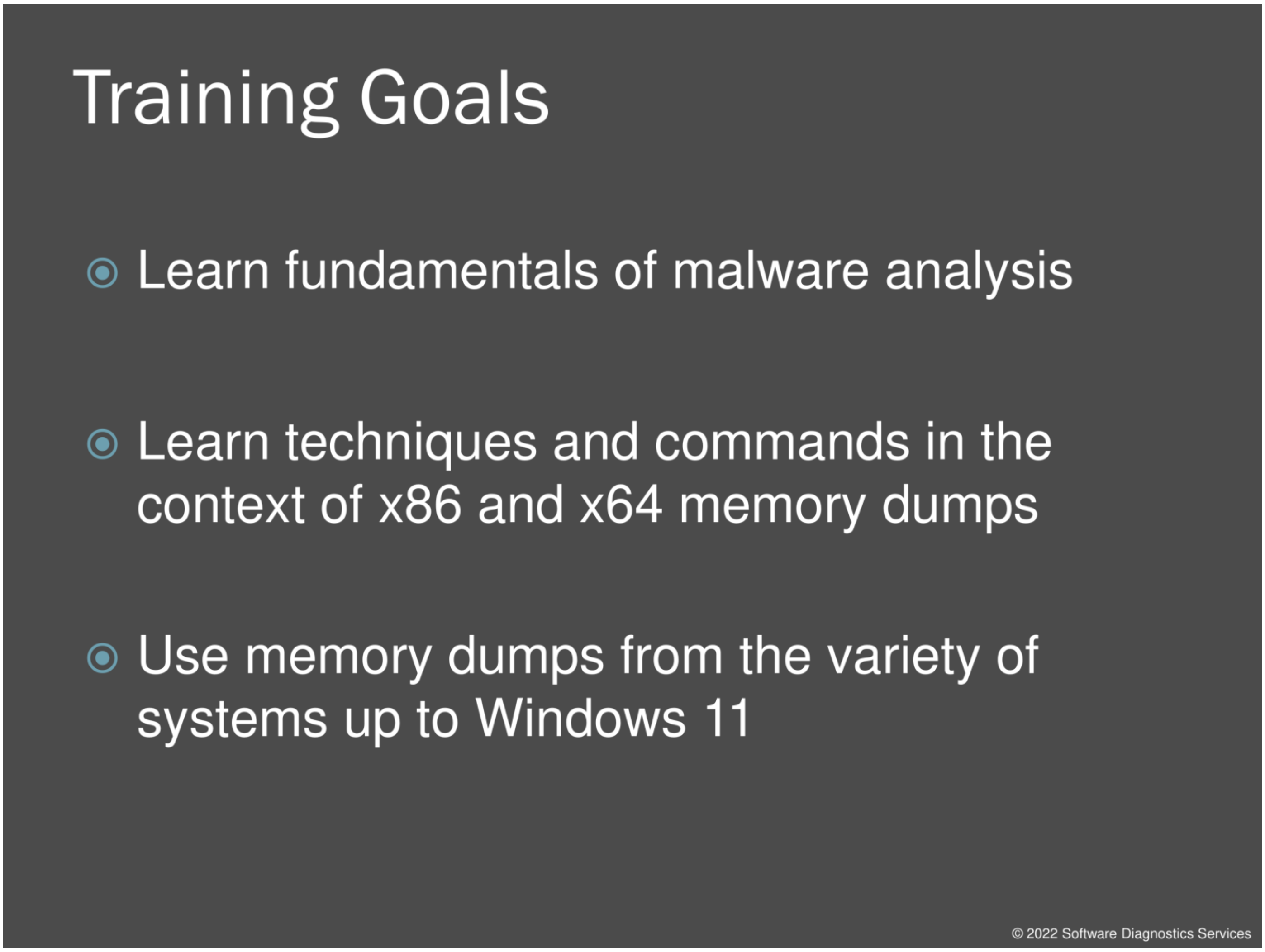


Prerequisites

Any of these:

- Basic and intermediate level Windows memory dump analysis using WinDbg
- C/C++/C# debugging skills
- Malware analysis (not WinDbg)

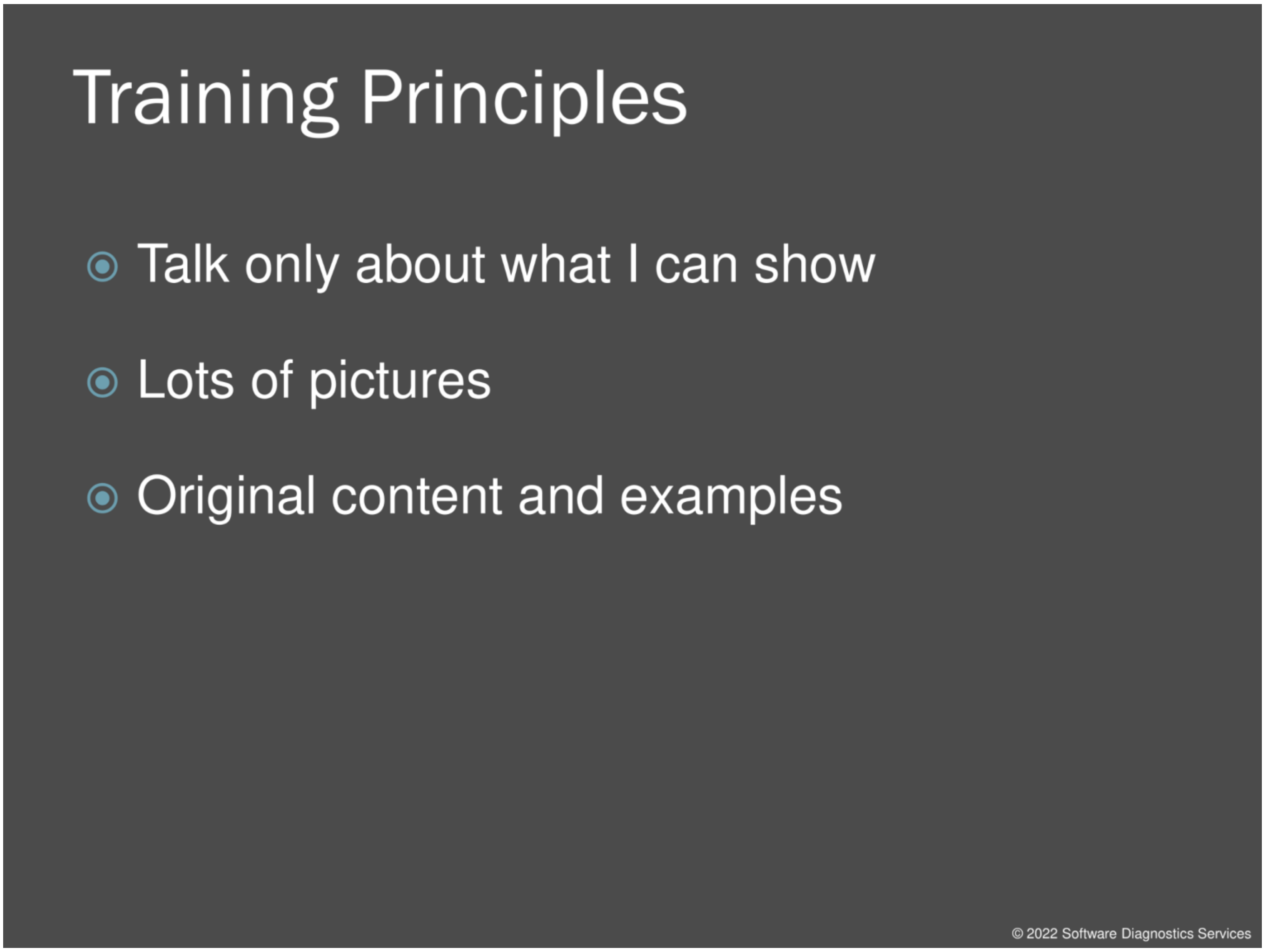
The main audience for this training is technical support and escalation engineers who analyze memory dumps from complex software environments using WinDbg debugger from Debugging Tools for Windows and need to check for possible malware presence in cases of abnormal software behavior. Software engineers, quality assurance and software maintenance engineers, security researchers, malware and memory forensics analysts who have never used this WinDbg debugger for analysis of computer memory may find this training useful as well as they learn how familiar malware detection and analysis concepts map into WinDbg commands. The ability to read assembly language has some advantages but is not strictly necessary.



Training Goals

- Learn fundamentals of malware analysis
- Learn techniques and commands in the context of x86 and x64 memory dumps
- Use memory dumps from the variety of systems up to Windows 11

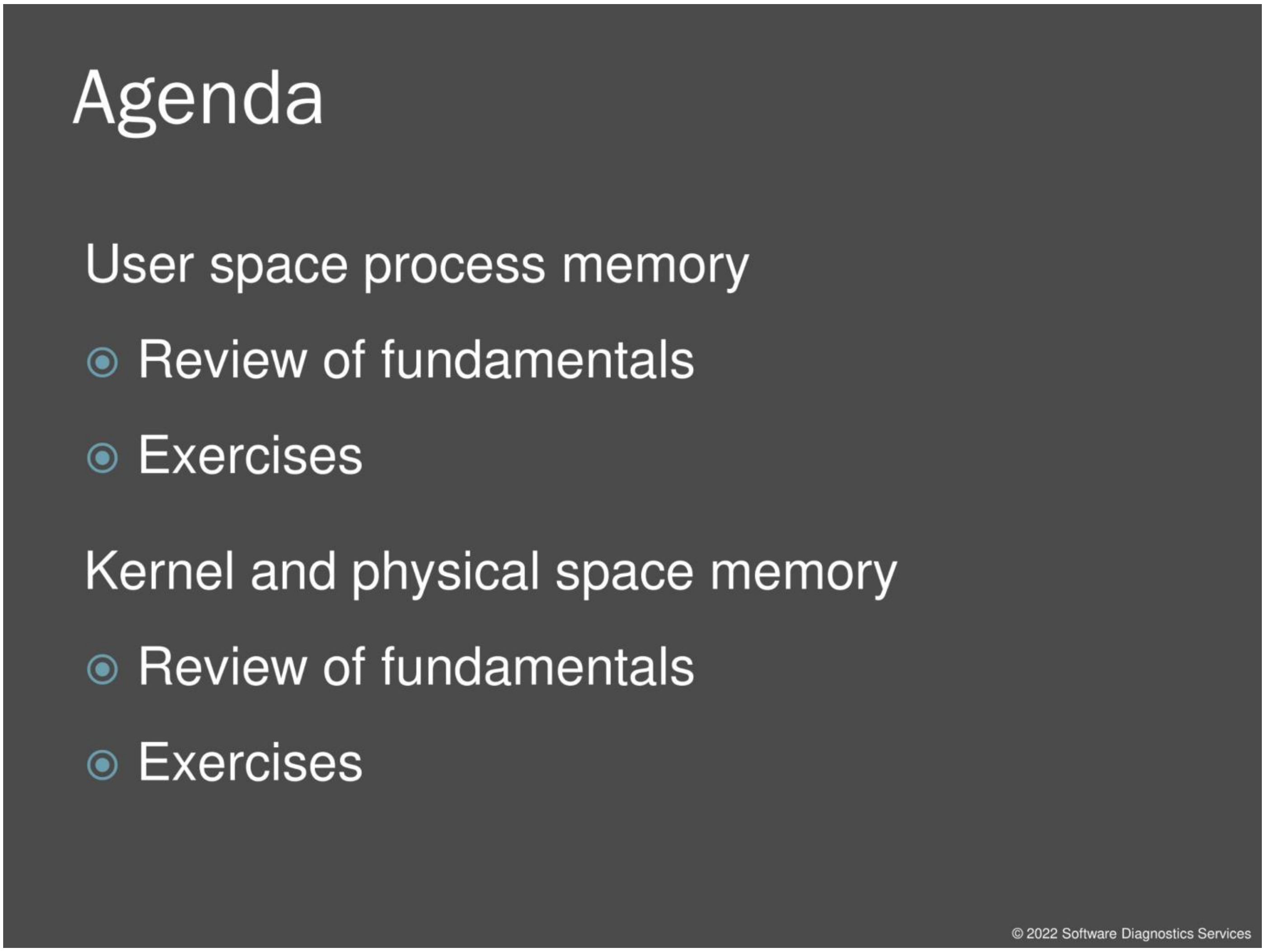
Our primary goal is to learn malware memory dump analysis in an accelerated fashion. In other accelerated courses, we first reviewed absolutely essential fundamentals necessary for memory dump analysis. Here we decided to review them as needed and start with analysis after a few introductory slides. During this course, we learn how to analyze different types of memory dumps such as process, kernel, and complete or physical memory. Kernel minidumps are not covered in this training because they are similar to kernel memory dumps with much less information saved available for analysis, and we need to be very lucky to find traces of malware in minidumps. Also, this training is about memory dump analysis and not about memory dump collection methods, tricks, and tips, although I provide you with a reference for memory acquisition during this training.



Training Principles

- Talk only about what I can show
- Lots of pictures
- Original content and examples

For me, there were many training formats to consider for this training, and I decided that the best way is to concentrate on exercises and explain concepts as necessary because the main audience should be familiar with WinDbg already.



Agenda

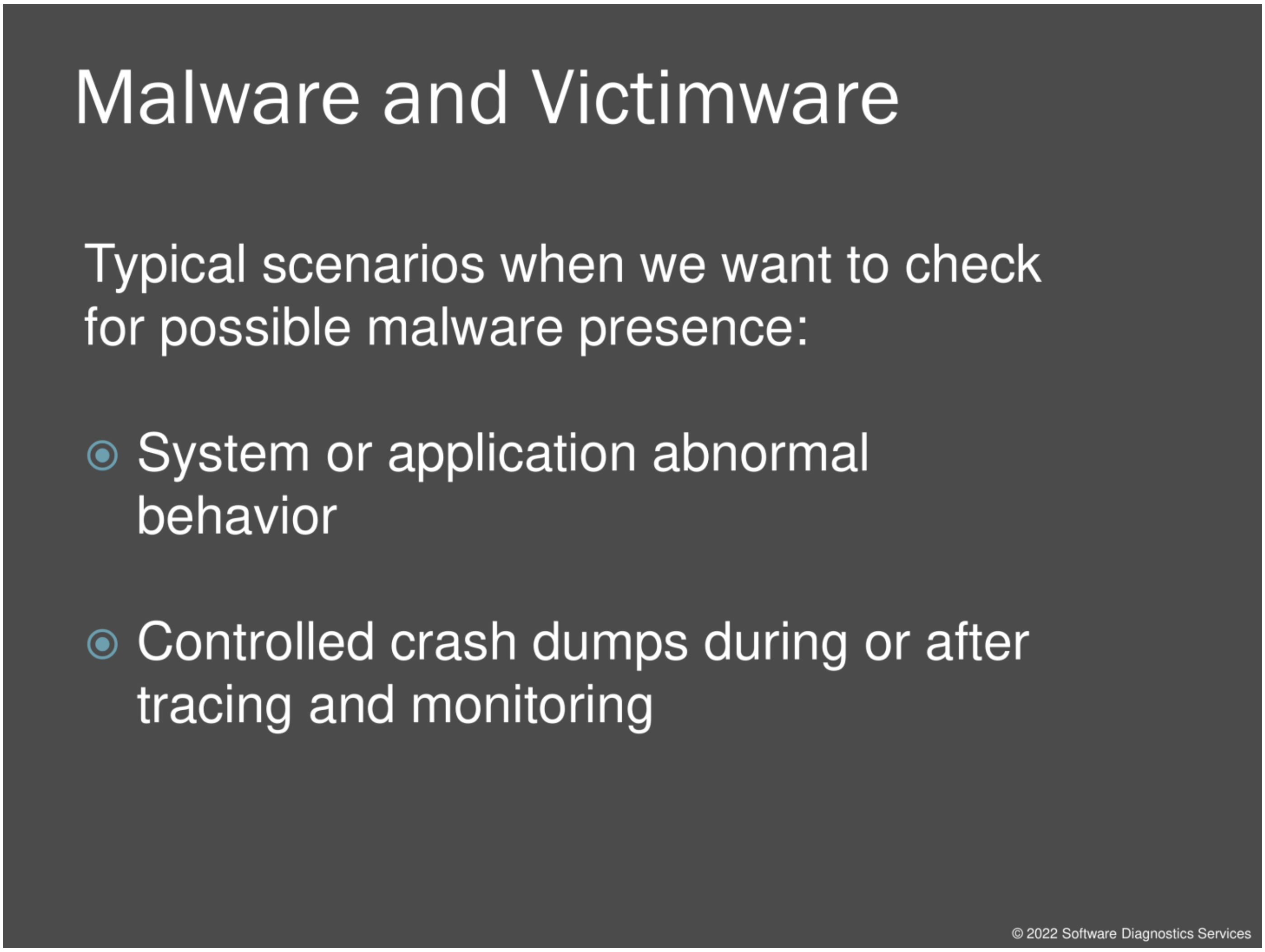
User space process memory

- Review of fundamentals
- Exercises

Kernel and physical space memory

- Review of fundamentals
- Exercises

This course is split into two parts: user space process memory analysis and kernel and complete or physical space analysis.

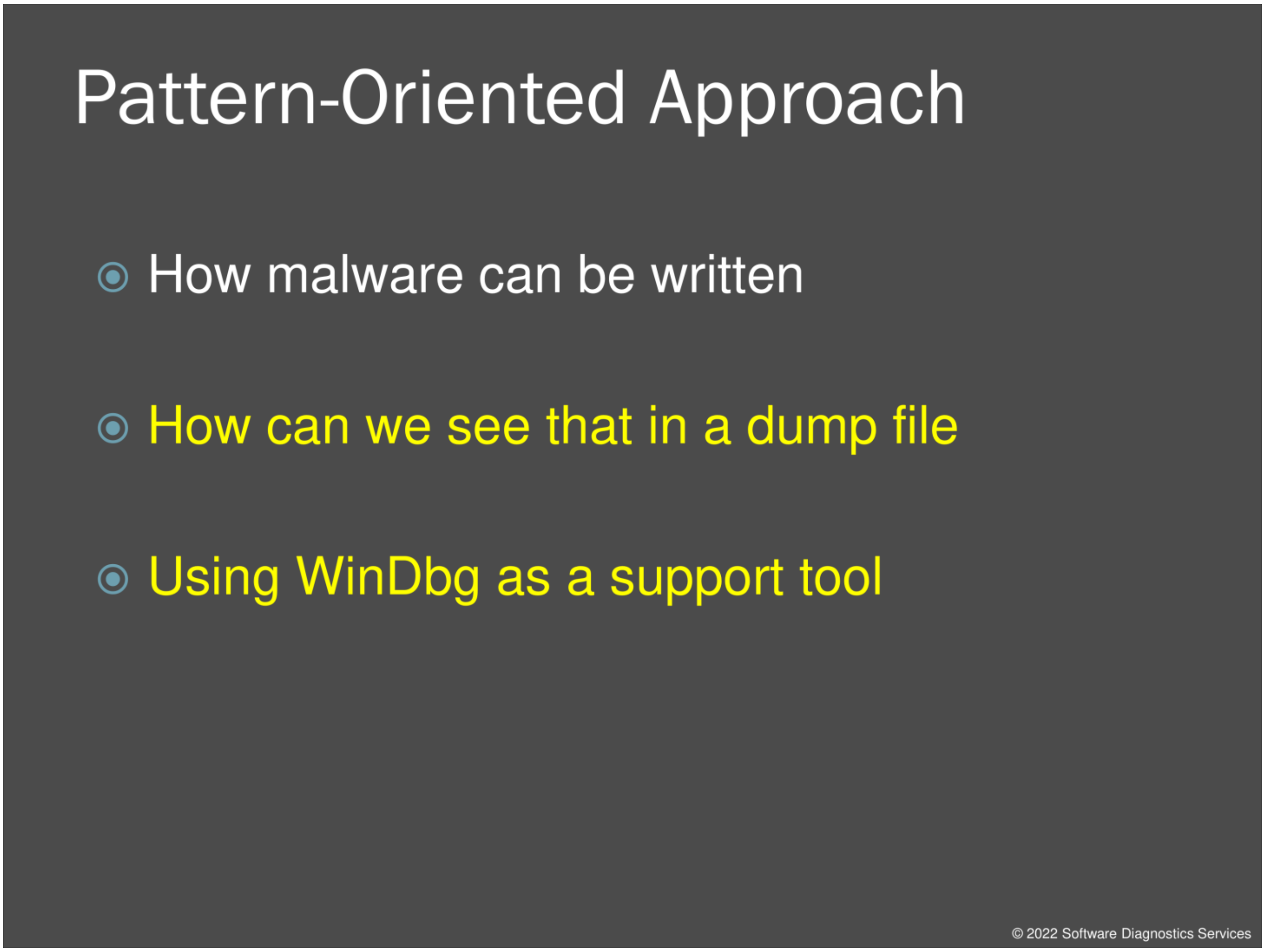


Malware and Victimware

Typical scenarios when we want to check for possible malware presence:

- System or application abnormal behavior
- Controlled crash dumps during or after tracing and monitoring

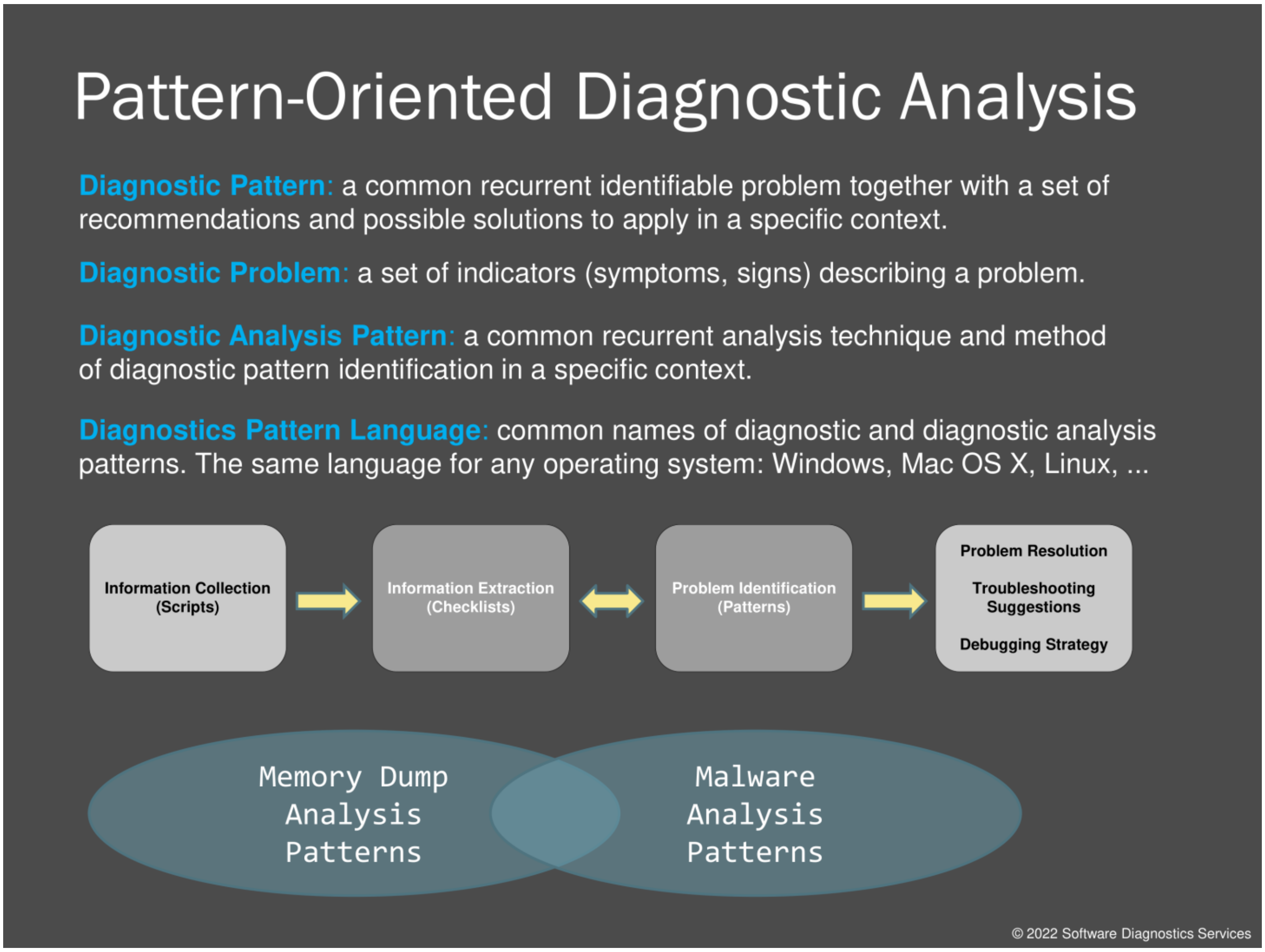
Because this course is primarily targeted to support engineers, there are typical scenarios when we want to check for possible malware presence. First, there are typical situations when we have abnormal software behavior such as crashes, hangs, CPU spikes, and memory leaks. All these can result not only from unintentional software defects or complex component interaction but also from malware mistakes and could also result from the intentional shutdown of processes and systems (some sort of denial-of-service attacks). The second scenario is when we proactively seek memory dump analysis or analyze memory dumps as supplemental artifacts to accompany software traces and logs. Note that malware may be completely transparent to observed software behavior that can be the same without malware.



Pattern-Oriented Approach

- How malware can be written
- How can we see that in a dump file
- Using WinDbg as a support tool

Here we outline our approach based on the main audience of this training. From our analysis of how malware can be written, we show through practical exercises how we can see that in memory dump files using WinDbg Preview or WinDbg debugger from Debugging Tools for Windows. This tool is a primary support tool for analyzing computer memory in Windows software support teams.



Pattern-Oriented Diagnostic Analysis

Diagnostic Pattern: a common recurrent identifiable problem together with a set of recommendations and possible solutions to apply in a specific context.

Diagnostic Problem: a set of indicators (symptoms, signs) describing a problem.

Diagnostic Analysis Pattern: a common recurrent analysis technique and method of diagnostic pattern identification in a specific context.

Diagnostics Pattern Language: common names of diagnostic and diagnostic analysis patterns. The same language for any operating system: Windows, Mac OS X, Linux, ...

Information Collection (Scripts) → Information Extraction (Checklists) ↔ Problem Identification (Patterns) → Problem Resolution

Troubleshooting Suggestions

Debugging Strategy

Memory Dump Analysis Patterns

Malware Analysis Patterns

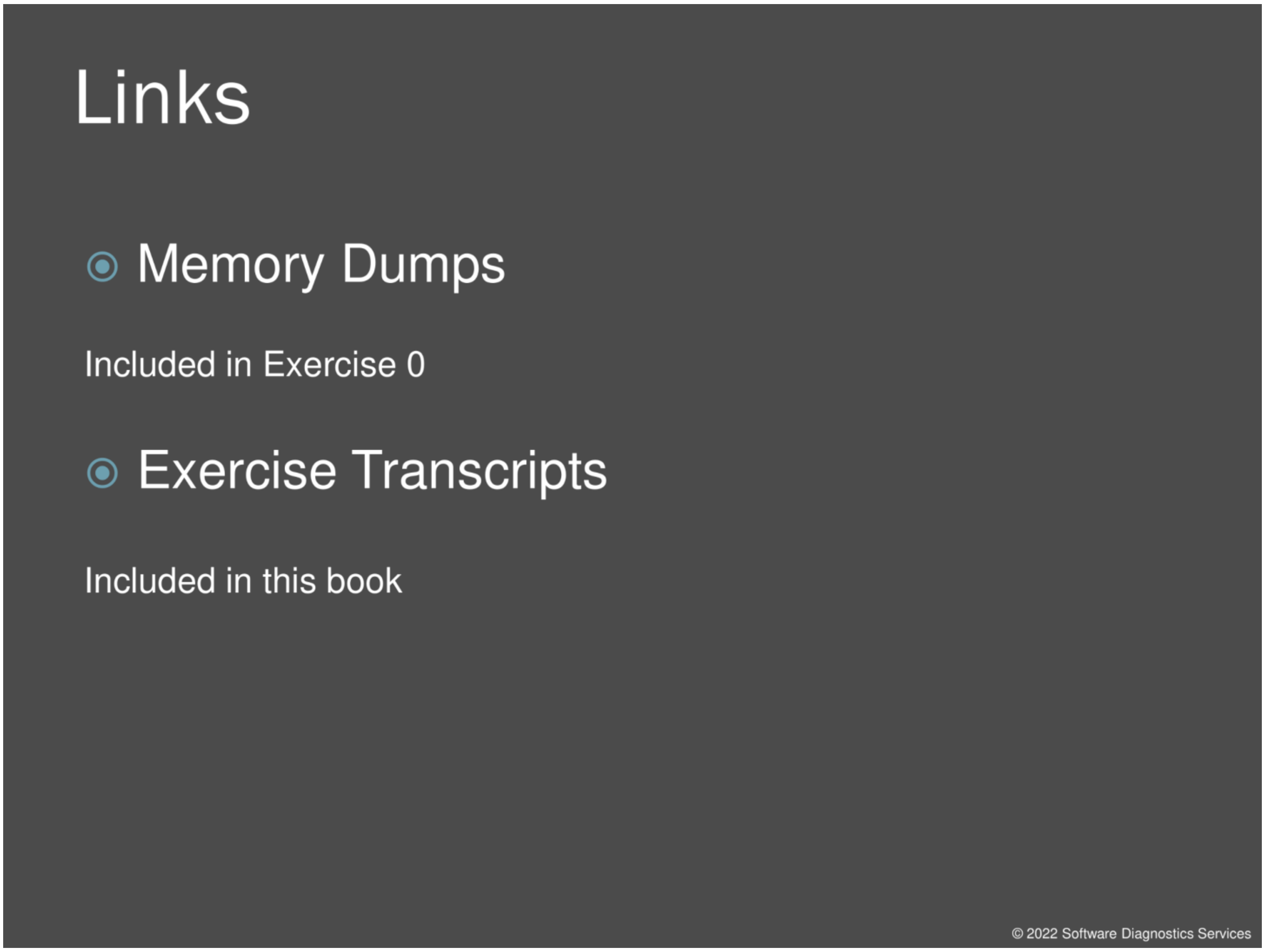
© 2022 Software Diagnostics Services

A few words about logs, checklists, and patterns. Memory dump analysis is usually an analysis of a text for the presence of patterns. We run commands, they output text, and then we look at that textual output, and when we find something suspicious, we execute more commands. Here checklists can be very useful. We provide a checklist by the end of this training. In some cases (such as complete memory dumps), it is beneficial to collect information into one log file by running several commands at once (like a script) and then do the first-order analysis. We do that during our complete memory dump analysis exercise. Malware analysis patterns are patterns of intentional abnormal structure and behavior. Because signs of non-intentional behavior and intentional non-malicious behavior such as value-adding hooking and code patching may be the same as intentional malicious behavior, such patterns may overlap with memory dump analysis patterns.

Practice Exercises

Practice Exercises

Now we come to practice. The goal is to show you important commands and how their output helps recognize malware analysis patterns.



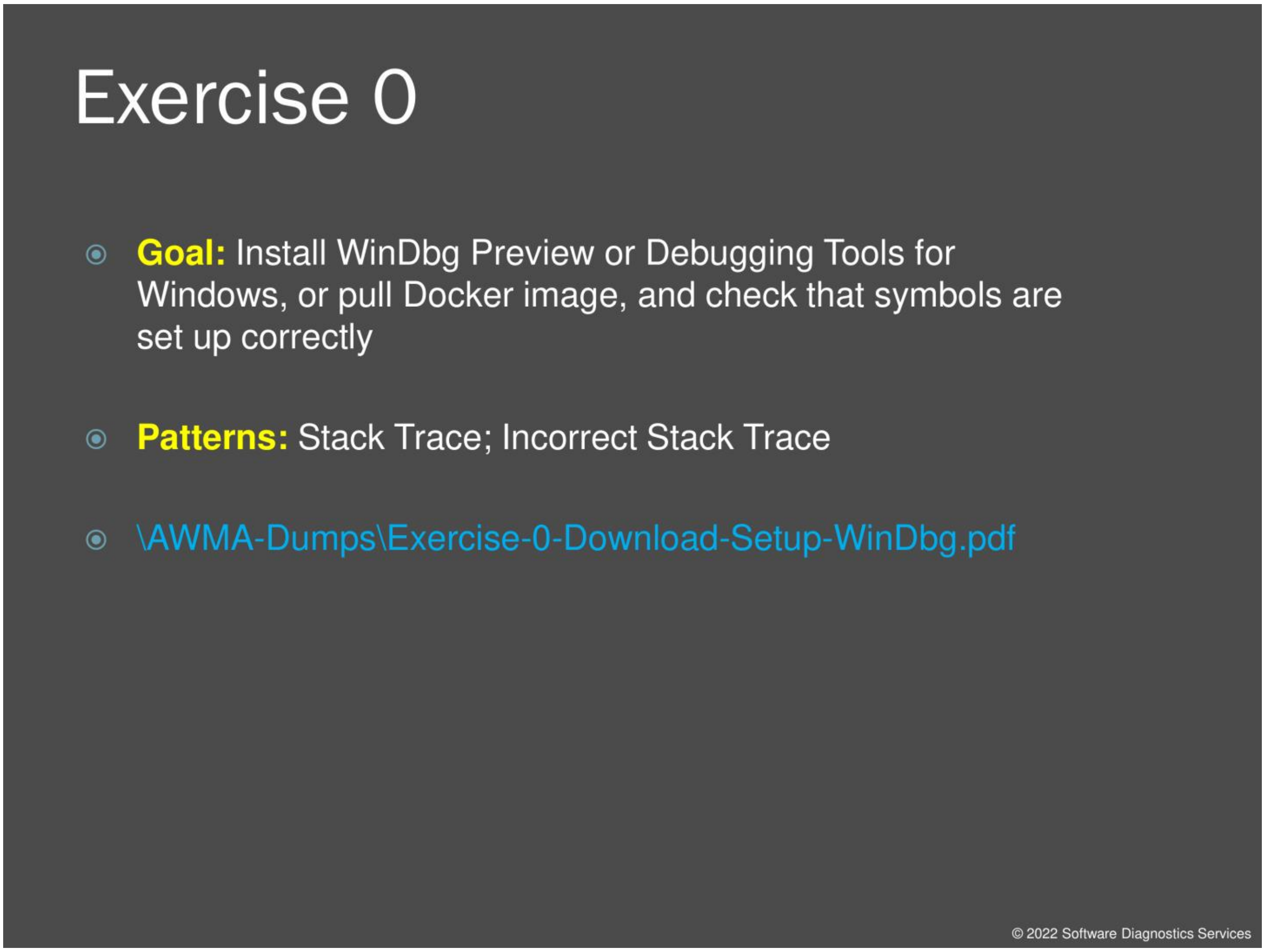
Links

- Memory Dumps

Included in Exercise 0

- Exercise Transcripts

Included in this book



Exercise 0

- **Goal:** Install WinDbg Preview or Debugging Tools for Windows, or pull Docker image, and check that symbols are set up correctly
- **Patterns:** Stack Trace; Incorrect Stack Trace
- \AWMA-Dumps\Exercise-0-Download-Setup-WinDbg.pdf

Here I assume you already prepared the environment, and I skip this exercise.

Exercise 0: Download, setup, and verify your WinDbg Preview or WinDbg installation, or Docker Debugging Tools for Windows image

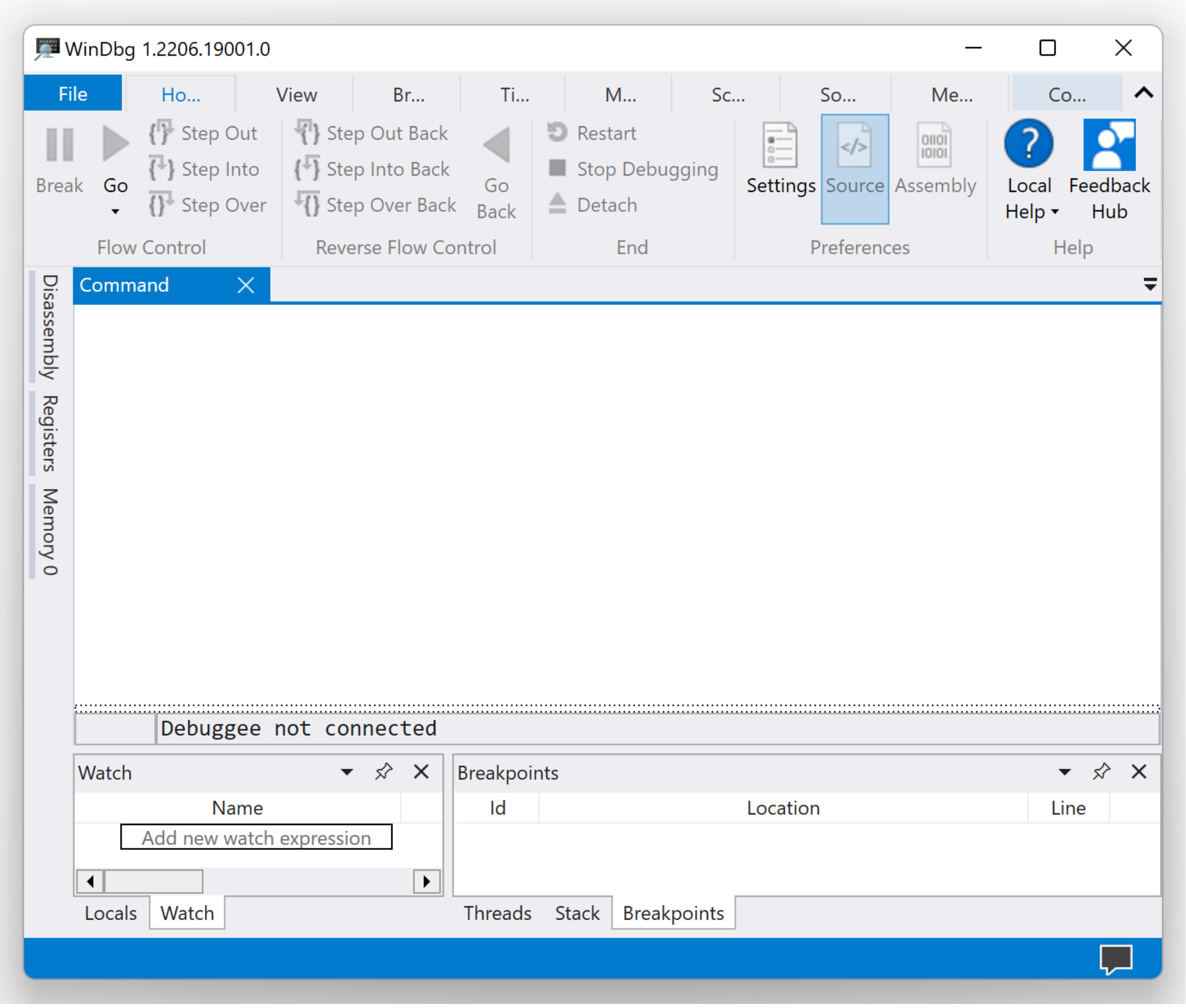
Goal: Install WinDbg Preview or Debugging Tools for Windows, or pull Docker image, and check that symbols are set up correctly.

Patterns: Stack Trace; Incorrect Stack Trace.

1. Download memory dump files if you haven't done that already and unpack the archive:

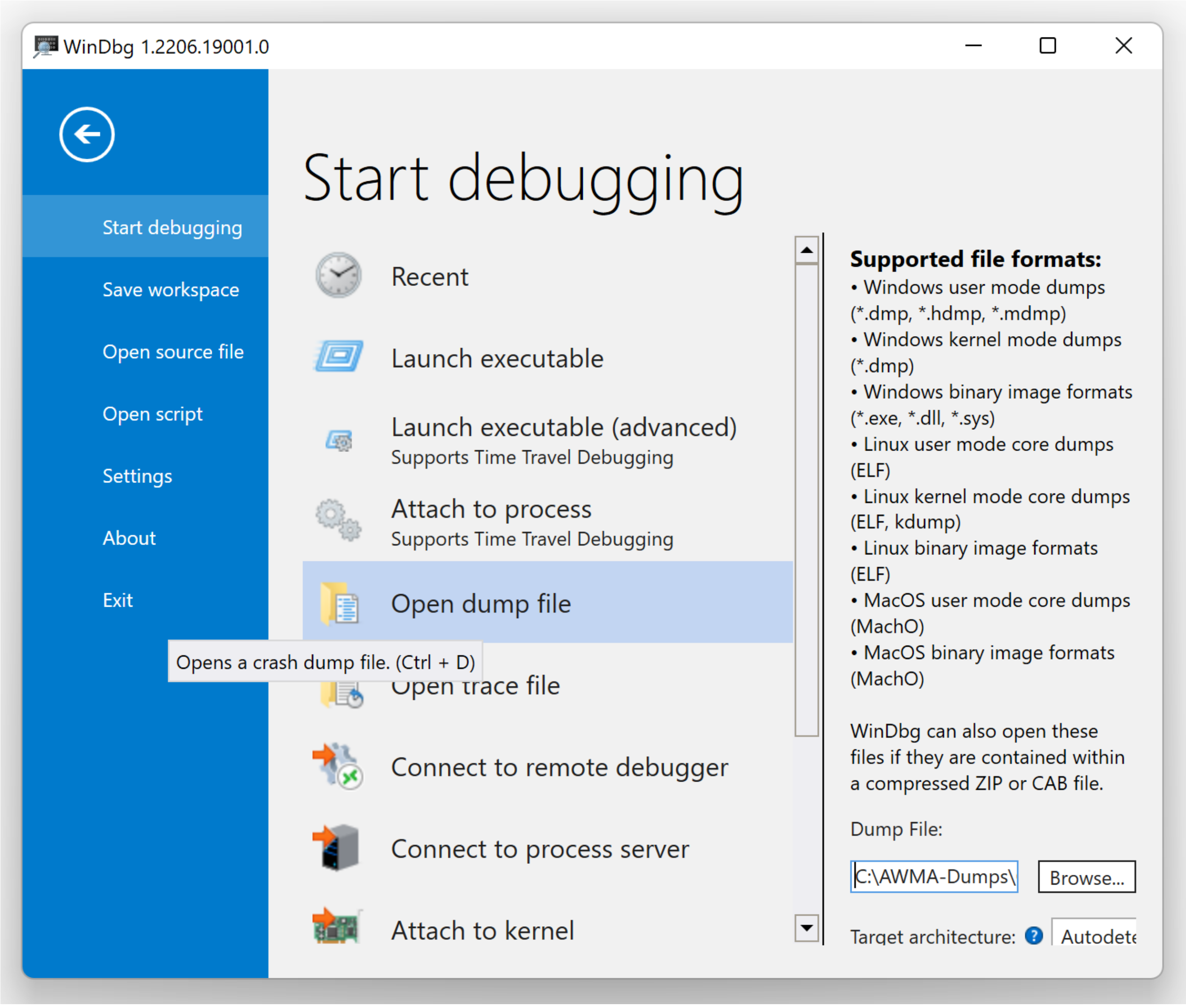
```
https://www.patterndiagnostics.com/Training/AWMA/AWMA3-Dumps-Part1.zip
https://www.patterndiagnostics.com/Training/AWMA/AWMA-Dumps-Part2.zip
https://www.patterndiagnostics.com/Training/AWMA/AWMA-Dumps-Part3.zip
https://www.patterndiagnostics.com/Training/AWMA/AWMA-Dumps-Part4.zip
https://www.patterndiagnostics.com/Training/AWMA/AWMA3-Dumps-Part5.zip
https://www.patterndiagnostics.com/Training/AWMA/InjectionResidue.zip
```

2. Install WinDbg Preview from Microsoft Store. Then, run the WinDbg Preview app.



```
Debuggee not connected
```

3. Open \AWMA-Dumps\Processes\wordpad.DMP:



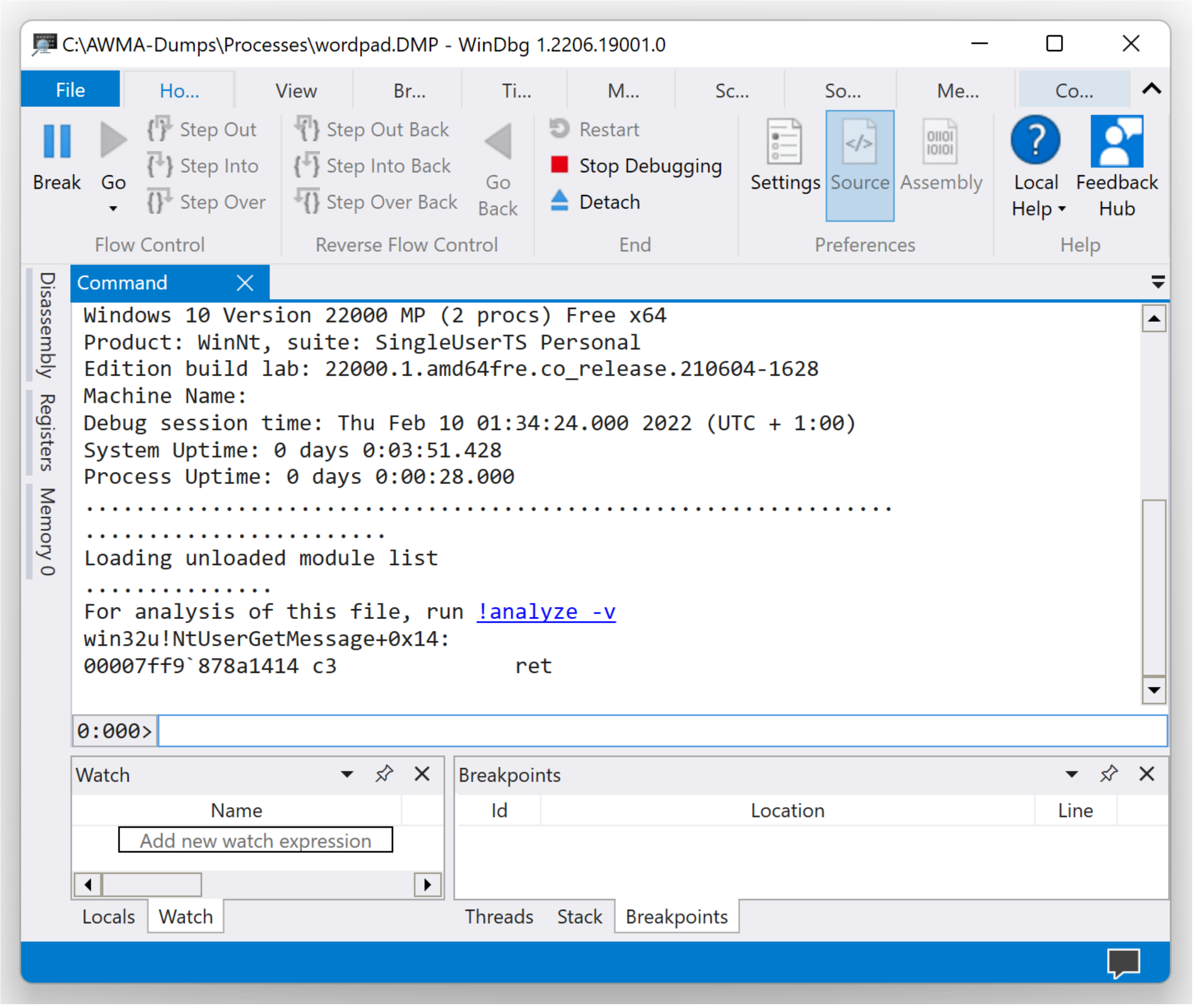
Supported file formats:
- Windows user mode dumps (*.dmp, *.hdmp, *.mdmp)
- Windows kernel mode dumps (*.dmp)
- Windows binary image formats (*.exe, *.dll, *.sys)
- Linux user mode core dumps (ELF)
- Linux kernel mode core dumps (ELF, kdump)
- Linux binary image formats (ELF)
- MacOS user mode core dumps (MachO)
- MacOS binary image formats (MachO)

WinDbg can also open these files if they are contained within a compressed ZIP or CAB file.

Dump File: C:\AWMA-Dumps\ Browse...

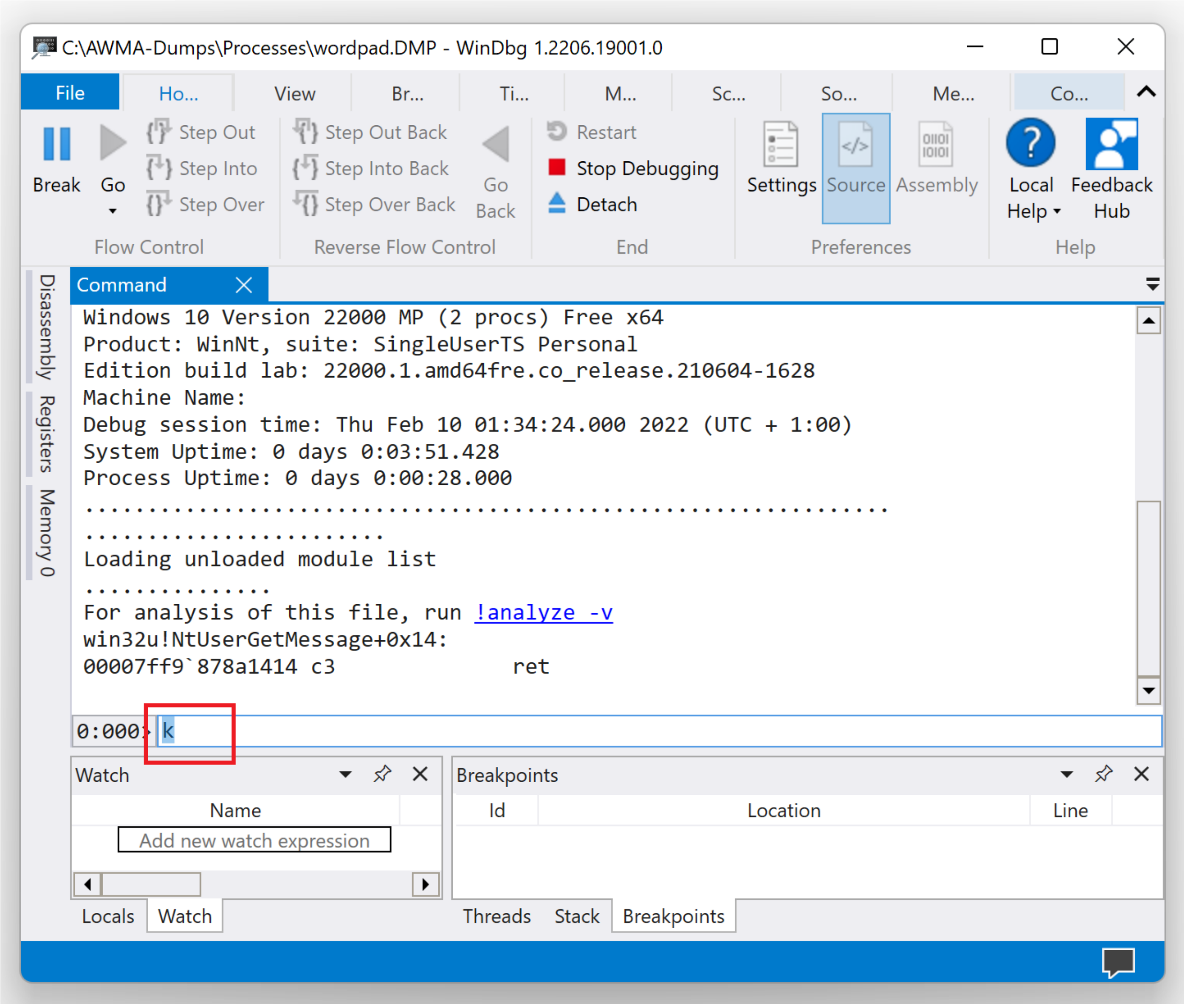
Target architecture: Autodete

4. We get the dump file loaded:



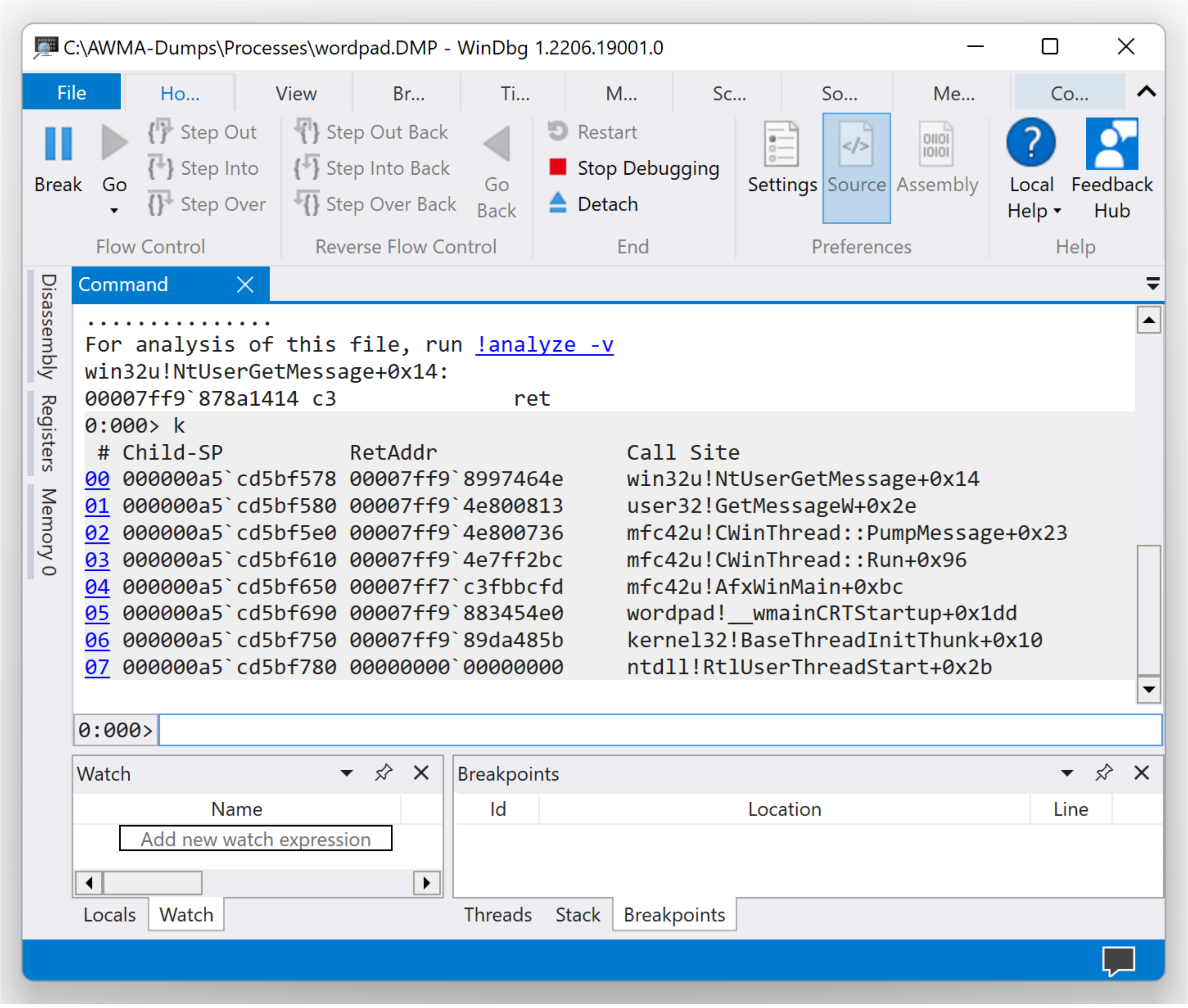
```
Windows 10 Version 22000 MP (2 procs) Free x64
Product: WinNt, suite: SingleUserTS Personal
Edition build lab: 22000.1.amd64fre.co_release.210604-1628
Machine Name:
Debug session time: Thu Feb 10 01:34:24.000 2022 (UTC + 1:00)
System Uptime: 0 days 0:03:51.428
Process Uptime: 0 days 0:00:28.000
...............................................................
........................
Loading unloaded module list
...............
For analysis of this file, run !analyze -v
win32u!NtUserGetMessage+0x14:
00007ff9`878a1414 c3              ret
0:000>
```

5. We can execute the **k** command to get the stack trace:



```
Windows 10 Version 22000 MP (2 procs) Free x64
Product: WinNt, suite: SingleUserTS Personal
Edition build lab: 22000.1.amd64fre.co_release.210604-1628
Machine Name:
Debug session time: Thu Feb 10 01:34:24.000 2022 (UTC + 1:00)
System Uptime: 0 days 0:03:51.428
Process Uptime: 0 days 0:00:28.000
...............................................................
........................
Loading unloaded module list
...............
For analysis of this file, run !analyze -v
win32u!NtUserGetMessage+0x14:
00007ff9`878a1414 c3              ret
```

```
0:000> k
```



```
...............
For analysis of this file, run !analyze -v
win32u!NtUserGetMessage+0x14:
00007ff9`878a1414 c3              ret
0:000> k
 # Child-SP          RetAddr               Call Site
00 000000a5`cd5bf578 00007ff9`8997464e     win32u!NtUserGetMessage+0x14
01 000000a5`cd5bf580 00007ff9`4e800813     user32!GetMessageW+0x2e
02 000000a5`cd5bf5e0 00007ff9`4e800736     mfc42u!CWinThread::PumpMessage+0x23
03 000000a5`cd5bf610 00007ff9`4e7ff2bc     mfc42u!CWinThread::Run+0x96
04 000000a5`cd5bf650 00007ff7`c3fbbcfd     mfc42u!AfxWinMain+0xbc
05 000000a5`cd5bf690 00007ff9`883454e0     wordpad!__wmainCRTStartup+0x1dd
06 000000a5`cd5bf750 00007ff9`89da485b     kernel32!BaseThreadInitThunk+0x10
07 000000a5`cd5bf780 00000000`00000000     ntdll!RtlUserThreadStart+0x2b
0:000>
```

6. The output of the **k** command should be this:

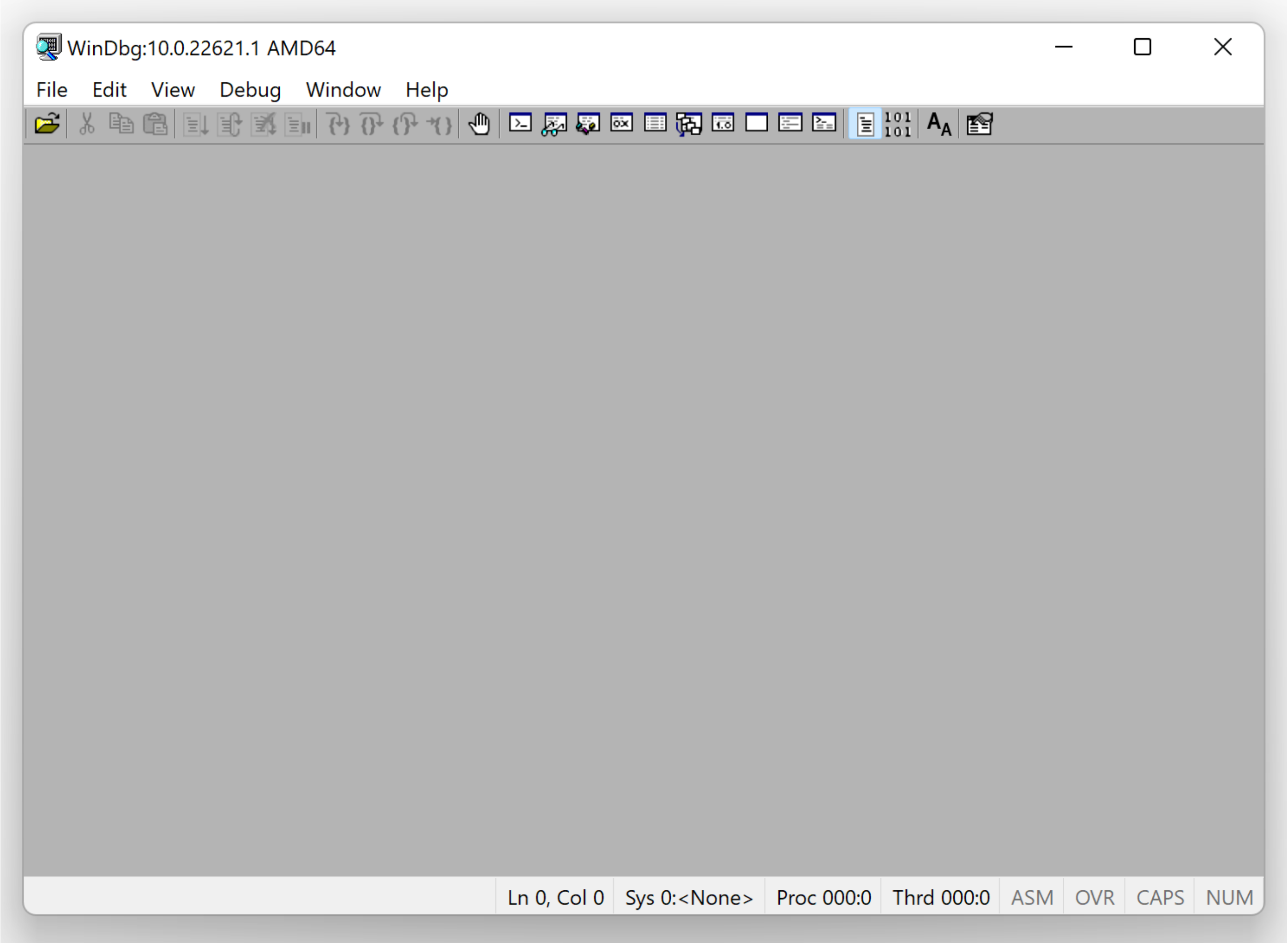
```
0:000> k
 # Child-SP          RetAddr               Call Site
00 000000a5`cd5bf578 00007ff9`8997464e     win32u!NtUserGetMessage+0x14
01 000000a5`cd5bf580 00007ff9`4e800813     user32!GetMessageW+0x2e
02 000000a5`cd5bf5e0 00007ff9`4e800736     mfc42u!CWinThread::PumpMessage+0x23
03 000000a5`cd5bf610 00007ff9`4e7ff2bc     mfc42u!CWinThread::Run+0x96
04 000000a5`cd5bf650 00007ff7`c3fbbcfd     mfc42u!AfxWinMain+0xbc
05 000000a5`cd5bf690 00007ff9`883454e0     wordpad!__wmainCRTStartup+0x1dd
06 000000a5`cd5bf750 00007ff9`89da485b     kernel32!BaseThreadInitThunk+0x10
07 000000a5`cd5bf780 00000000`00000000     ntdll!RtlUserThreadStart+0x2b
```

If it has this form below with a large offset, then your symbol files were not set up correctly – **Incorrect Stack Trace** pattern:

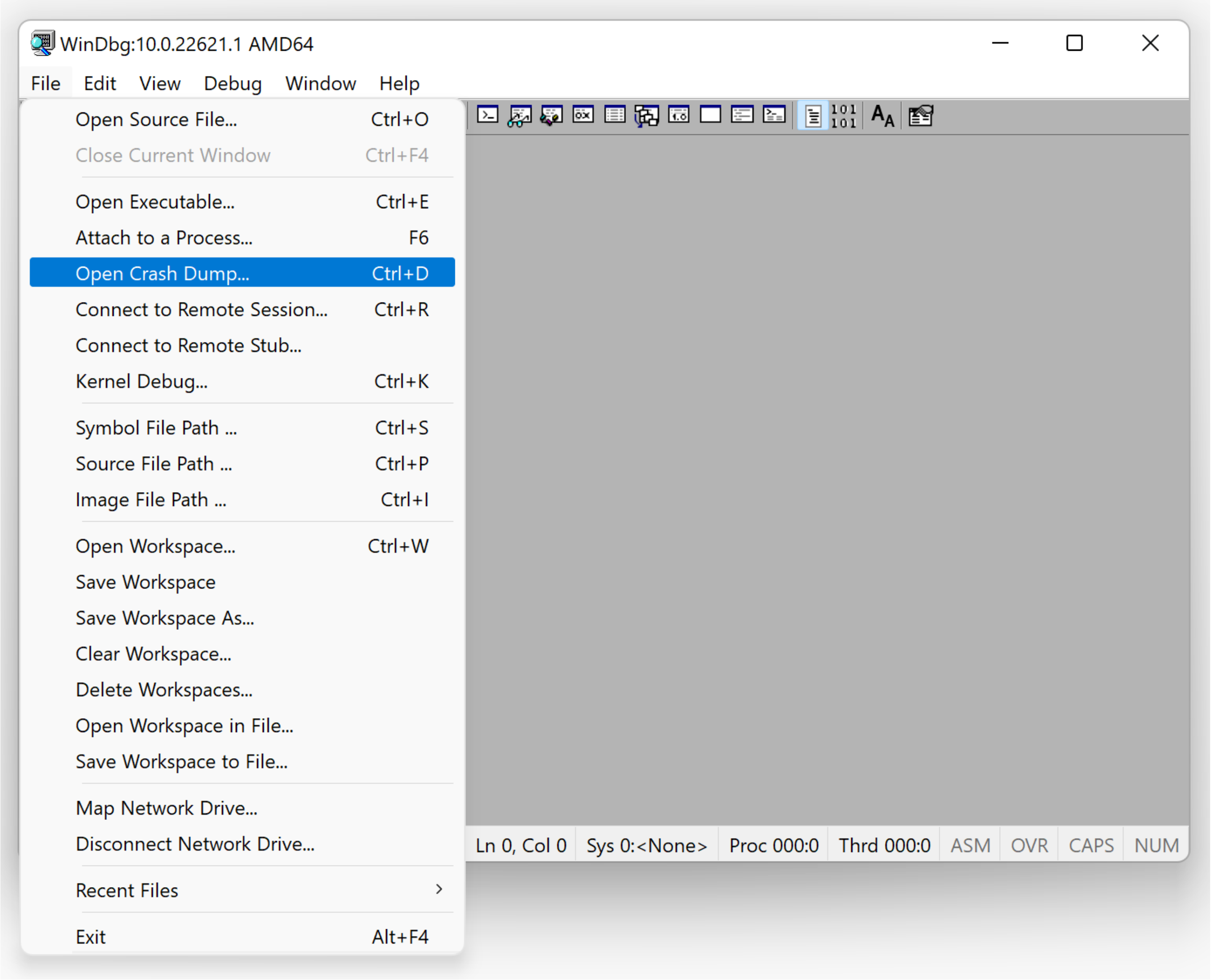
```
0:000> k
 # Child-SP          RetAddr               Call Site
00 000000a5`cd5bf578 00007ff9`8997464e     win32u!NtUserGetMessage+0x14
01 000000a5`cd5bf580 00007ff9`4e800813     user32!GetMessageW+0x2e
02 000000a5`cd5bf5e0 00007ff9`4e800736     mfc42u!Ordinal5730+0x23
03 000000a5`cd5bf610 00007ff9`4e7ff2bc     mfc42u!Ordinal6054+0x96
04 000000a5`cd5bf650 00007ff7`c3fbbcfd     mfc42u!Ordinal1584+0xbc
05 000000a5`cd5bf690 00007ff9`883454e0     wordpad+0xbcfd
06 000000a5`cd5bf750 00007ff9`89da485b     kernel32!BaseThreadInitThunk+0x10
07 000000a5`cd5bf780 00000000`00000000     ntdll!RtlUserThreadStart+0x2b
```

7. [Optional] Download and install the recommended version of Debugging Tools for Windows (See windbg.org for quick links, WinDbg Quick Links \ Download Debugging Tools for Windows). For this part, we use WinDbg 10.0.22621.1 from Windows SDK 10.0.22621 for Windows 11, version 22H2.

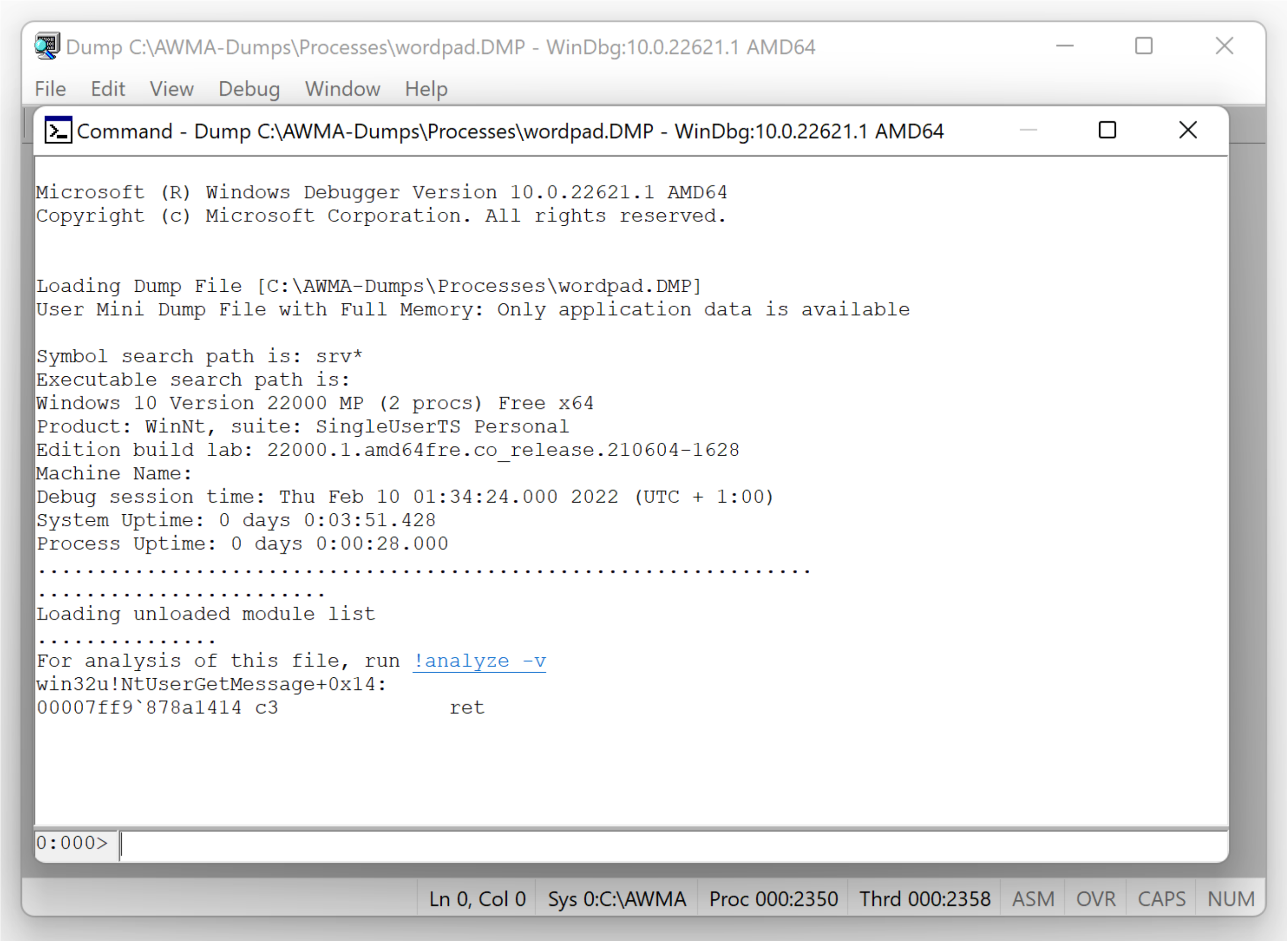
8. Launch WinDbg from Windows Kits \ WinDbg (X64).



9. Open \AWMA-Dumps\Processes\wordpad.DMP:

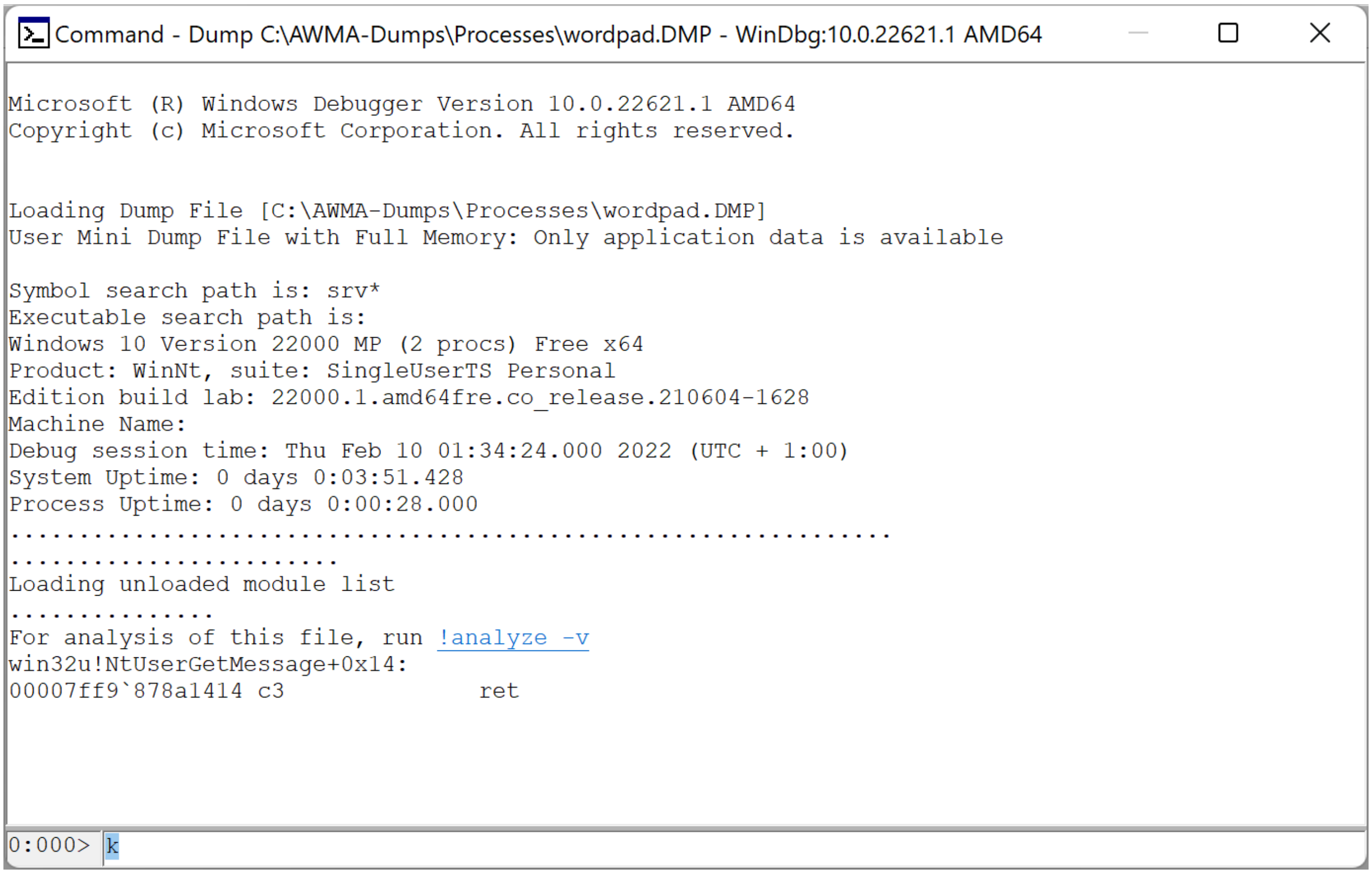

WinDbg:10.0.22621.1 AMD64
File
Edit
View
Debug
Window
Help
Open Source File... Ctrl+O
Close Current Window Ctrl+F4
Open Executable... Ctrl+E
Attach to a Process... F6
Open Crash Dump... Ctrl+D
Connect to Remote Session... Ctrl+R
Connect to Remote Stub...
Kernel Debug... Ctrl+K
Symbol File Path ... Ctrl+S
Source File Path ... Ctrl+P
Image File Path ... Ctrl+I
Open Workspace... Ctrl+W
Save Workspace
Save Workspace As...
Clear Workspace...
Delete Workspaces...
Open Workspace in File...
Save Workspace to File...
Map Network Drive...
Disconnect Network Drive...
Recent Files
Exit Alt+F4
Ln 0, Col 0
Sys 0:<None>
Proc 000:0
Thrd 000:0
ASM
OVR
CAPS
NUM

10. We get the dump file loaded:


Dump C:\AWMA-Dumps\Processes\wordpad.DMP - WinDbg:10.0.22621.1 AMD64
File Edit View Debug Window Help
Command - Dump C:\AWMA-Dumps\Processes\wordpad.DMP - WinDbg:10.0.22621.1 AMD64
Ln 0, Col 0
Sys 0:C:\AWMA
Proc 000:2350
Thrd 000:2358
ASM
OVR
CAPS
NUM

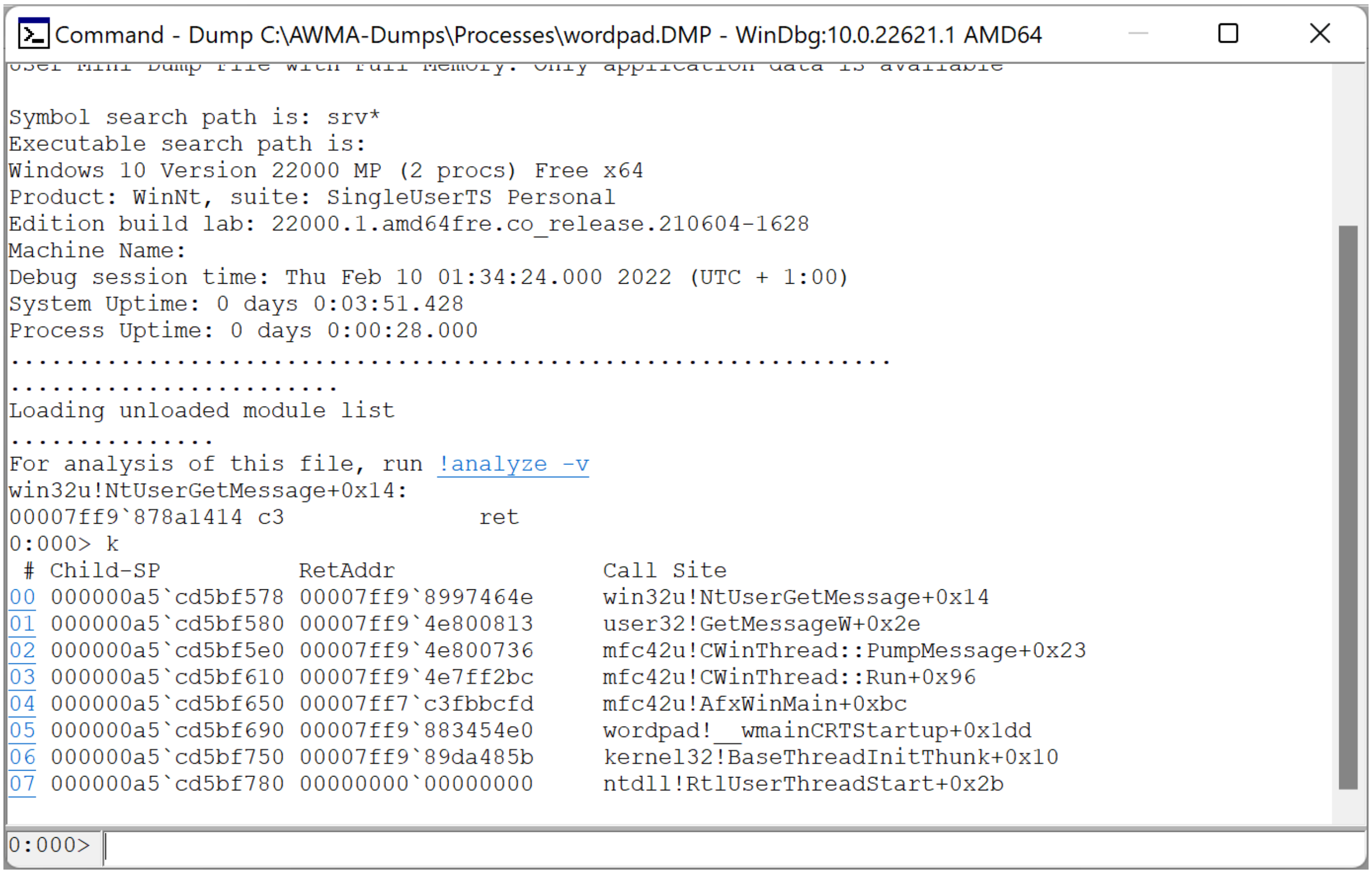
```
Microsoft (R) Windows Debugger Version 10.0.22621.1 AMD64
Copyright (c) Microsoft Corporation. All rights reserved.


Loading Dump File [C:\AWMA-Dumps\Processes\wordpad.DMP]
User Mini Dump File with Full Memory: Only application data is available

Symbol search path is: srv*
Executable search path is:
Windows 10 Version 22000 MP (2 procs) Free x64
Product: WinNt, suite: SingleUserTS Personal
Edition build lab: 22000.1.amd64fre.co_release.210604-1628
Machine Name:
Debug session time: Thu Feb 10 01:34:24.000 2022 (UTC + 1:00)
System Uptime: 0 days 0:03:51.428
Process Uptime: 0 days 0:00:28.000
................................................................
........................
Loading unloaded module list
...............
For analysis of this file, run !analyze -v
win32u!NtUserGetMessage+0x14:
00007ff9`878a1414 c3              ret

0:000>
```

11. Type **k** command to verify the correctness of stack trace:



Command - Dump C:\AWMA-Dumps\Processes\wordpad.DMP - WinDbg:10.0.22621.1 AMD64

```
Microsoft (R) Windows Debugger Version 10.0.22621.1 AMD64
Copyright (c) Microsoft Corporation. All rights reserved.


Loading Dump File [C:\AWMA-Dumps\Processes\wordpad.DMP]
User Mini Dump File with Full Memory: Only application data is available

Symbol search path is: srv*
Executable search path is:
Windows 10 Version 22000 MP (2 procs) Free x64
Product: WinNt, suite: SingleUserTS Personal
Edition build lab: 22000.1.amd64fre.co_release.210604-1628
Machine Name:
Debug session time: Thu Feb 10 01:34:24.000 2022 (UTC + 1:00)
System Uptime: 0 days 0:03:51.428
Process Uptime: 0 days 0:00:28.000
................................................................
........................
Loading unloaded module list
...............
For analysis of this file, run !analyze -v
win32u!NtUserGetMessage+0x14:
00007ff9`878a1414 c3              ret

0:000> k
```



Command - Dump C:\AWMA-Dumps\Processes\wordpad.DMP - WinDbg:10.0.22621.1 AMD64

```
User Mini Dump File with Full Memory: Only application data is available

Symbol search path is: srv*
Executable search path is:
Windows 10 Version 22000 MP (2 procs) Free x64
Product: WinNt, suite: SingleUserTS Personal
Edition build lab: 22000.1.amd64fre.co_release.210604-1628
Machine Name:
Debug session time: Thu Feb 10 01:34:24.000 2022 (UTC + 1:00)
System Uptime: 0 days 0:03:51.428
Process Uptime: 0 days 0:00:28.000
................................................................
........................
Loading unloaded module list
...............
For analysis of this file, run !analyze -v
win32u!NtUserGetMessage+0x14:
00007ff9`878a1414 c3              ret
0:000> k
 # Child-SP          RetAddr               Call Site
00 000000a5`cd5bf578 00007ff9`8997464e     win32u!NtUserGetMessage+0x14
01 000000a5`cd5bf580 00007ff9`4e800813     user32!GetMessageW+0x2e
02 000000a5`cd5bf5e0 00007ff9`4e800736     mfc42u!CWinThread::PumpMessage+0x23
03 000000a5`cd5bf610 00007ff9`4e7ff2bc     mfc42u!CWinThread::Run+0x96
04 000000a5`cd5bf650 00007ff7`c3fbbcfd     mfc42u!AfxWinMain+0xbc
05 000000a5`cd5bf690 00007ff9`883454e0     wordpad!__wmainCRTStartup+0x1dd
06 000000a5`cd5bf750 00007ff9`89da485b     kernel32!BaseThreadInitThunk+0x10
07 000000a5`cd5bf780 00000000`00000000     ntdll!RtlUserThreadStart+0x2b

0:000>
```

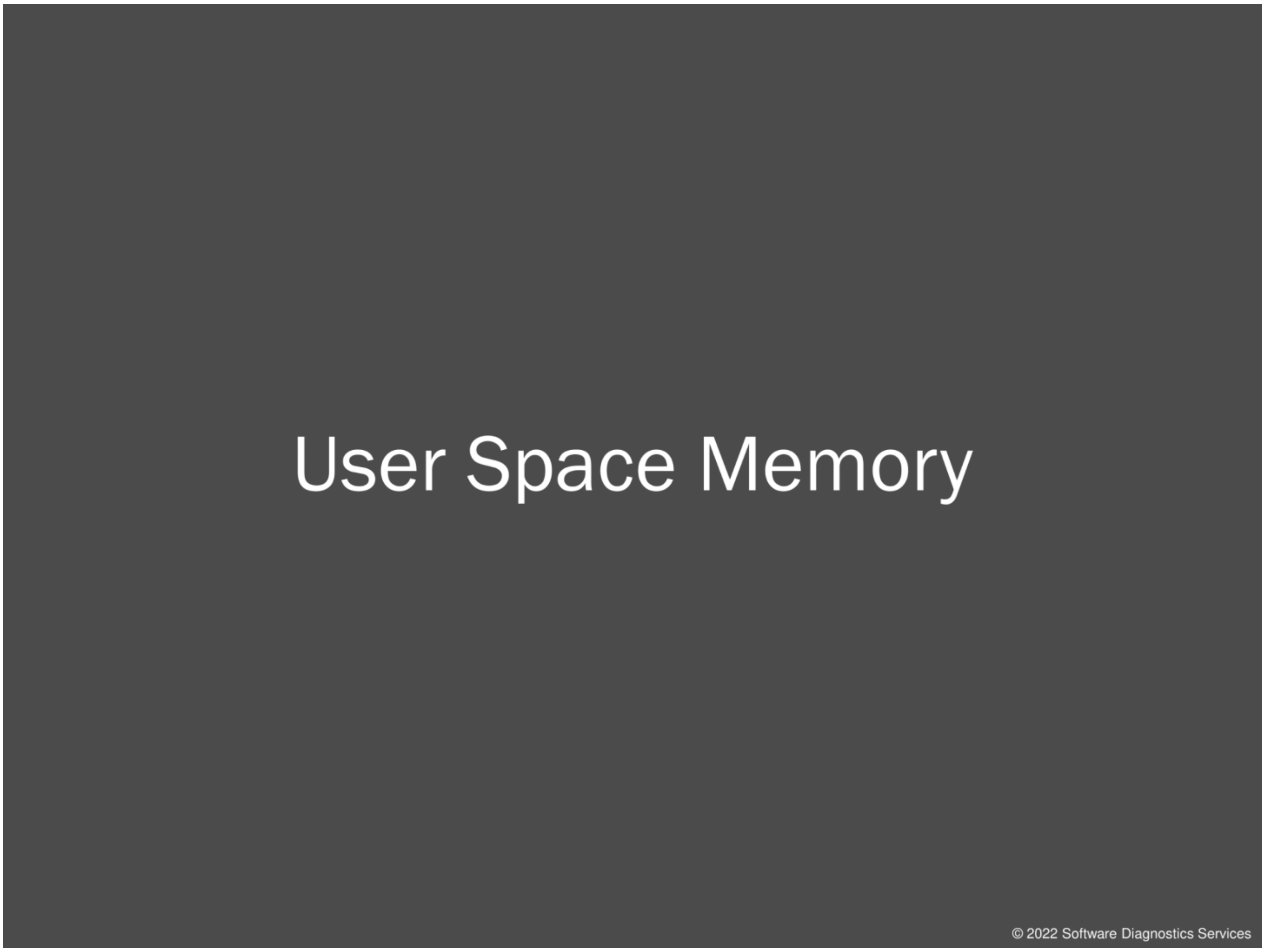
12. [Optional] Another approach is to use Docker container image that contains preinstalled WinDbg x64 with required symbol files for this course’s memory dump files:

```
C:\AWMA-Dumps>docker pull patterndiagnostics/windbg:10.0.25136.1001-awma

C:\AWMA-Dumps>docker run -it -v C:\AWMA-Dumps:C:\AWMA-Dumps
patterndiagnostics/windbg:10.0.25136.1001-awma

Microsoft Windows [Version 10.0.20348.768]
(c) Microsoft Corporation. All rights reserved.

C:\WinDbg>windbg C:\AWMA-Dumps\Processes\wordpad.DMP

Microsoft (R) Windows Debugger Version 10.0.22621.1 AMD64
Copyright (c) Microsoft Corporation. All rights reserved.


Loading Dump File [C:\AWMA-Dumps\Processes\wordpad.DMP]
User Mini Dump File with Full Memory: Only application data is available


************* Path validation summary **************
Response                         Time (ms)     Location
OK                                             C:\WinDbg\mss
Symbol search path is: C:\WinDbg\mss
Executable search path is:
Windows 10 Version 22000 MP (2 procs) Free x64
Product: WinNt, suite: SingleUserTS Personal
Edition build lab: 22000.1.amd64fre.co_release.210604-1628
Machine Name:
Debug session time: Thu Feb 10 01:34:24.000 2022 (UTC + 1:00)
System Uptime: 0 days 0:03:51.428
Process Uptime: 0 days 0:00:28.000
................................................................
........................
Loading unloaded module list
...............
For analysis of this file, run !analyze -v
win32u!NtUserGetMessage+0x14:
00007ff9`878a1414 c3              ret

0:000> k
Child-SP          RetAddr               Call Site
000000a5`cd5bf578 00007ff9`8997464e     win32u!NtUserGetMessage+0x14
000000a5`cd5bf580 00007ff9`4e800813     user32!GetMessageW+0x2e
000000a5`cd5bf5e0 00007ff9`4e800736     mfc42u!CWinThread::PumpMessage+0x23
000000a5`cd5bf610 00007ff9`4e7ff2bc     mfc42u!CWinThread::Run+0x96
000000a5`cd5bf650 00007ff7`c3fbbcfd     mfc42u!AfxWinMain+0xbc
000000a5`cd5bf690 00007ff9`883454e0     wordpad!__wmainCRTStartup+0x1dd
000000a5`cd5bf750 00007ff9`89da485b     kernel32!BaseThreadInitThunk+0x10
000000a5`cd5bf780 00000000`00000000     ntdll!RtlUserThreadStart+0x2b

0:000> q
quit:
NatVis script unloaded from 'C:\Program Files\Windows
Kits\10\Debuggers\x64\Visualizers\atlmfc.natvis'
NatVis script unloaded from 'C:\Program Files\Windows
Kits\10\Debuggers\x64\Visualizers\ObjectiveC.natvis'
NatVis script unloaded from 'C:\Program Files\Windows
Kits\10\Debuggers\x64\Visualizers\concurrency.natvis'
```

```
NatVis script unloaded from 'C:\Program Files\Windows
Kits\10\Debuggers\x64\Visualizers\cpp_rest.natvis'
NatVis script unloaded from 'C:\Program Files\Windows
Kits\10\Debuggers\x64\Visualizers\stl.natvis'
NatVis script unloaded from 'C:\Program Files\Windows
Kits\10\Debuggers\x64\Visualizers\Windows.Data.Json.natvis'
NatVis script unloaded from 'C:\Program Files\Windows
Kits\10\Debuggers\x64\Visualizers\Windows.Devices.Geolocation.natvis'
NatVis script unloaded from 'C:\Program Files\Windows
Kits\10\Debuggers\x64\Visualizers\Windows.Devices.Sensors.natvis'
NatVis script unloaded from 'C:\Program Files\Windows
Kits\10\Debuggers\x64\Visualizers\Windows.Media.natvis'
NatVis script unloaded from 'C:\Program Files\Windows
Kits\10\Debuggers\x64\Visualizers\windows.natvis'
NatVis script unloaded from 'C:\Program Files\Windows
Kits\10\Debuggers\x64\Visualizers\winrt.natvis'

C:\WinDbg>exit

C:\AWMA-Dumps>
```

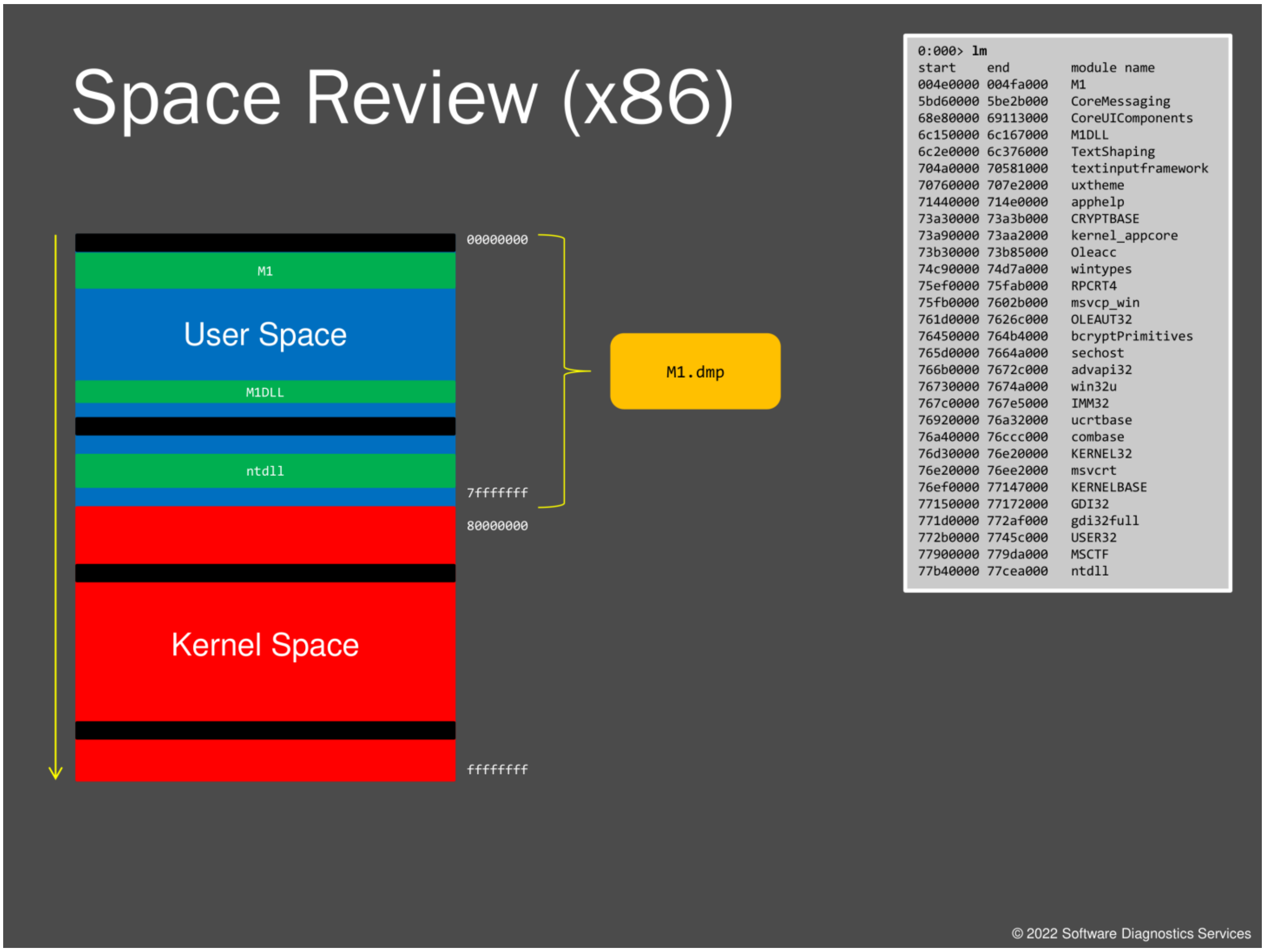
If you find any symbol problems, please use the Contact form on www.patterndiagnostics.com to report them.

We recommend exiting WinDbg or WinDbg Preview app after each exercise.



User Space Memory

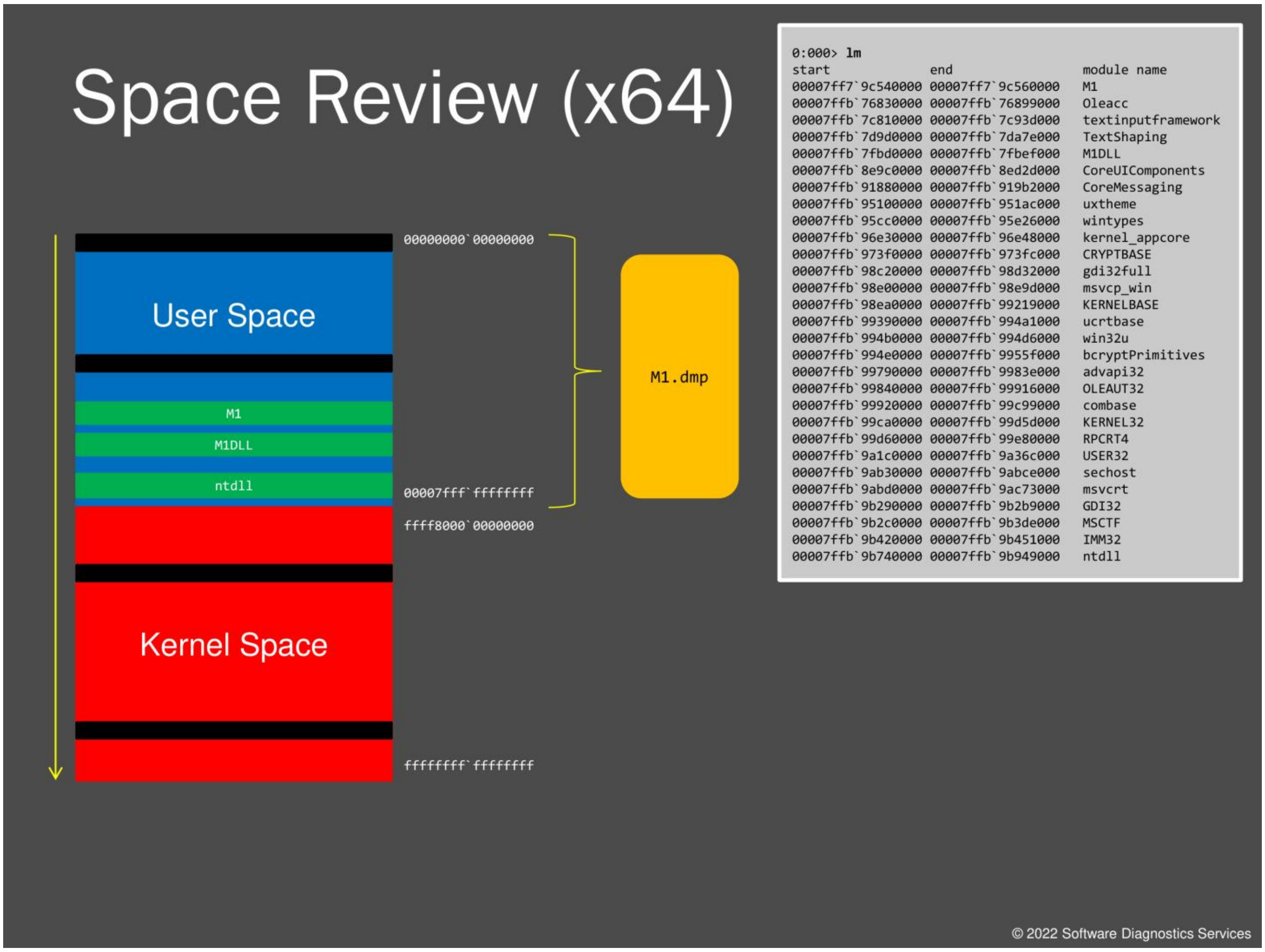
All exercises were modeled on real-life examples using specially constructed applications. All process memory dumps were saved from Windows Vista, Windows 7, and Windows 11 systems running on real hardware.



Space Review (x86)

```
0:000> lm
start    end        module name
004e0000 004fa000   M1
5bd60000 5be2b000   CoreMessaging
68e80000 69113000   CoreUIComponents
6c150000 6c167000   M1DLL
6c2e0000 6c376000   TextShaping
704a0000 70581000   textinputframework
70760000 707e2000   uxtheme
71440000 714e0000   apphelp
73a30000 73a3b000   CRYPTBASE
73a90000 73aa2000   kernel_appcore
73b30000 73b85000   Oleacc
74c90000 74d7a000   wintypes
75ef0000 75fab000   RPCRT4
75fb0000 7602b000   msvcp_win
761d0000 7626c000   OLEAUT32
76450000 764b4000   bcryptPrimitives
765d0000 7664a000   sechost
766b0000 7672c000   advapi32
76730000 7674a000   win32u
767c0000 767e5000   IMM32
76920000 76a32000   ucrtbase
76a40000 76ccc000   combase
76d30000 76e20000   KERNEL32
76e20000 76ee2000   msvcrt
76ef0000 77147000   KERNELBASE
77150000 77172000   GDI32
771d0000 772af000   gdi32full
772b0000 7745c000   USER32
77900000 779da000   MSCTF
77b40000 77cea000   ntdll
```

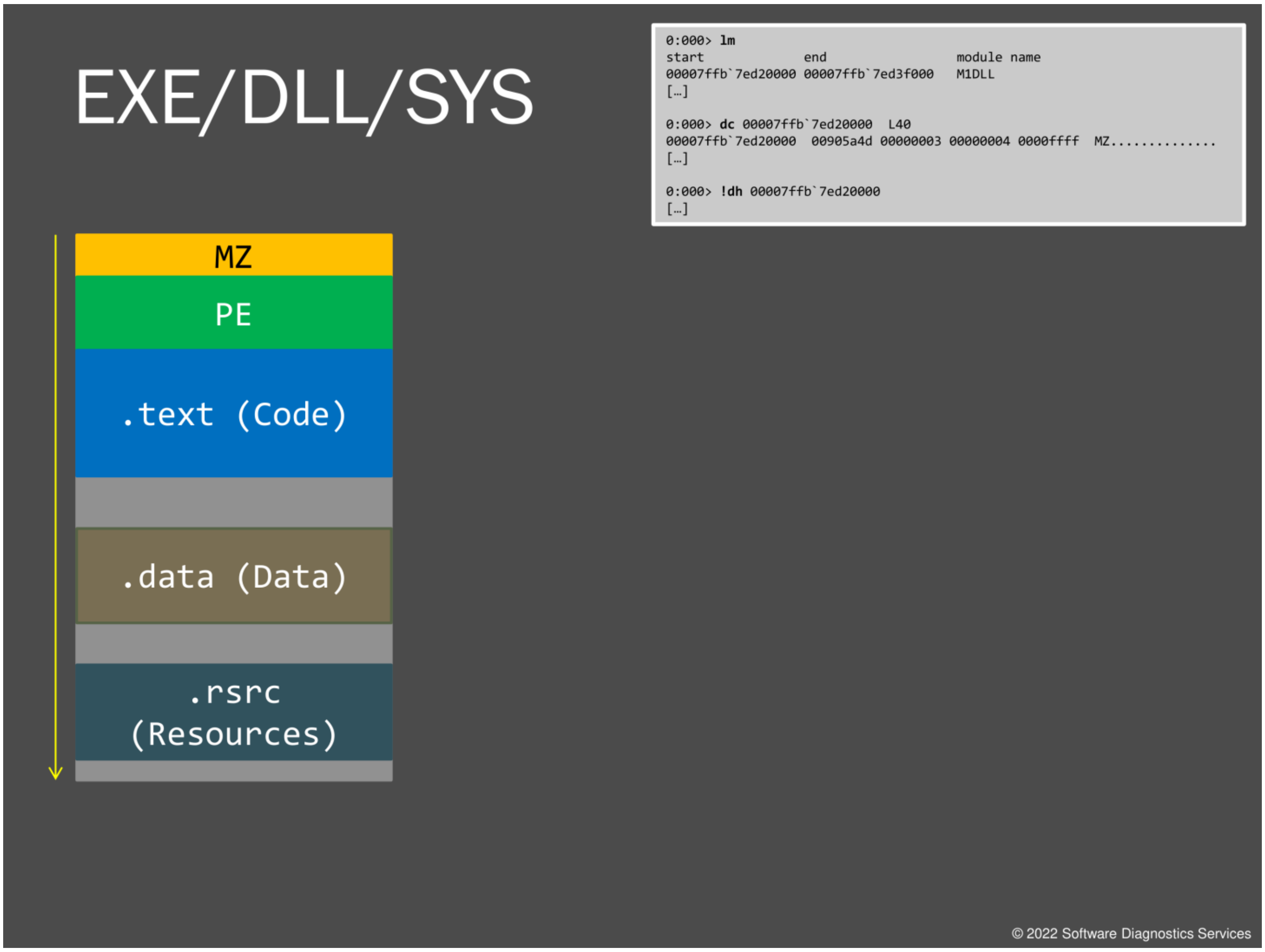
Most of you are familiar with the 32-bit process address space mapping. So I just briefly repeat that when we run an application or service, its executable file is loaded into memory, and if it references other DLLs, they are loaded too at some addresses in memory. There may be gaps between them, like black regions in this picture. Some memory is also allocated for additional working regions needed for process execution. What kind of memory is unimportant to us when we look at a process memory dump. It usually has a 2 GB range, and we see addresses where modules are loaded using the **lm** command. When we save a dump, all accessible memory, including loaded modules, is saved. The dump is usually much smaller than 2 GB unless we have a memory leak or an application is a memory demanding, such as an in-memory database. Please also note that we reversed the direction of the space diagram if you compare it with Accelerated Windows Memory Dump Analysis training to keep the same direction we see in WinDbg when we dump memory, such as when we have lower addresses on top and memory addresses increase down.



Space Review (x64)

```
0:000> lm
start             end                 module name
00007ff7`9c540000 00007ff7`9c560000   M1
00007ffb`76830000 00007ffb`76899000   Oleacc
00007ffb`7c810000 00007ffb`7c93d000   textinputframework
00007ffb`7d9d0000 00007ffb`7da7e000   TextShaping
00007ffb`7fbd0000 00007ffb`7fbef000   M1DLL
00007ffb`8e9c0000 00007ffb`8ed2d000   CoreUIComponents
00007ffb`91880000 00007ffb`919b2000   CoreMessaging
00007ffb`95100000 00007ffb`951ac000   uxtheme
00007ffb`95cc0000 00007ffb`95e26000   wintypes
00007ffb`96e30000 00007ffb`96e48000   kernel_appcore
00007ffb`973f0000 00007ffb`973fc000   CRYPTBASE
00007ffb`98c20000 00007ffb`98d32000   gdi32full
00007ffb`98e00000 00007ffb`98e9d000   msvcp_win
00007ffb`98ea0000 00007ffb`99219000   KERNELBASE
00007ffb`99390000 00007ffb`994a1000   ucrtbase
00007ffb`994b0000 00007ffb`994d6000   win32u
00007ffb`994e0000 00007ffb`9955f000   bcryptPrimitives
00007ffb`99790000 00007ffb`9983e000   advapi32
00007ffb`99840000 00007ffb`99916000   OLEAUT32
00007ffb`99920000 00007ffb`99c99000   combase
00007ffb`99ca0000 00007ffb`99d5d000   KERNEL32
00007ffb`99d60000 00007ffb`99e80000   RPCRT4
00007ffb`9a1c0000 00007ffb`9a36c000   USER32
00007ffb`9ab30000 00007ffb`9abce000   sechost
00007ffb`9abd0000 00007ffb`9ac73000   msvcrt
00007ffb`9b290000 00007ffb`9b2b9000   GDI32
00007ffb`9b2c0000 00007ffb`9b3de000   MSCTF
00007ffb`9b420000 00007ffb`9b451000   IMM32
00007ffb`9b740000 00007ffb`9b949000   ntdll
```

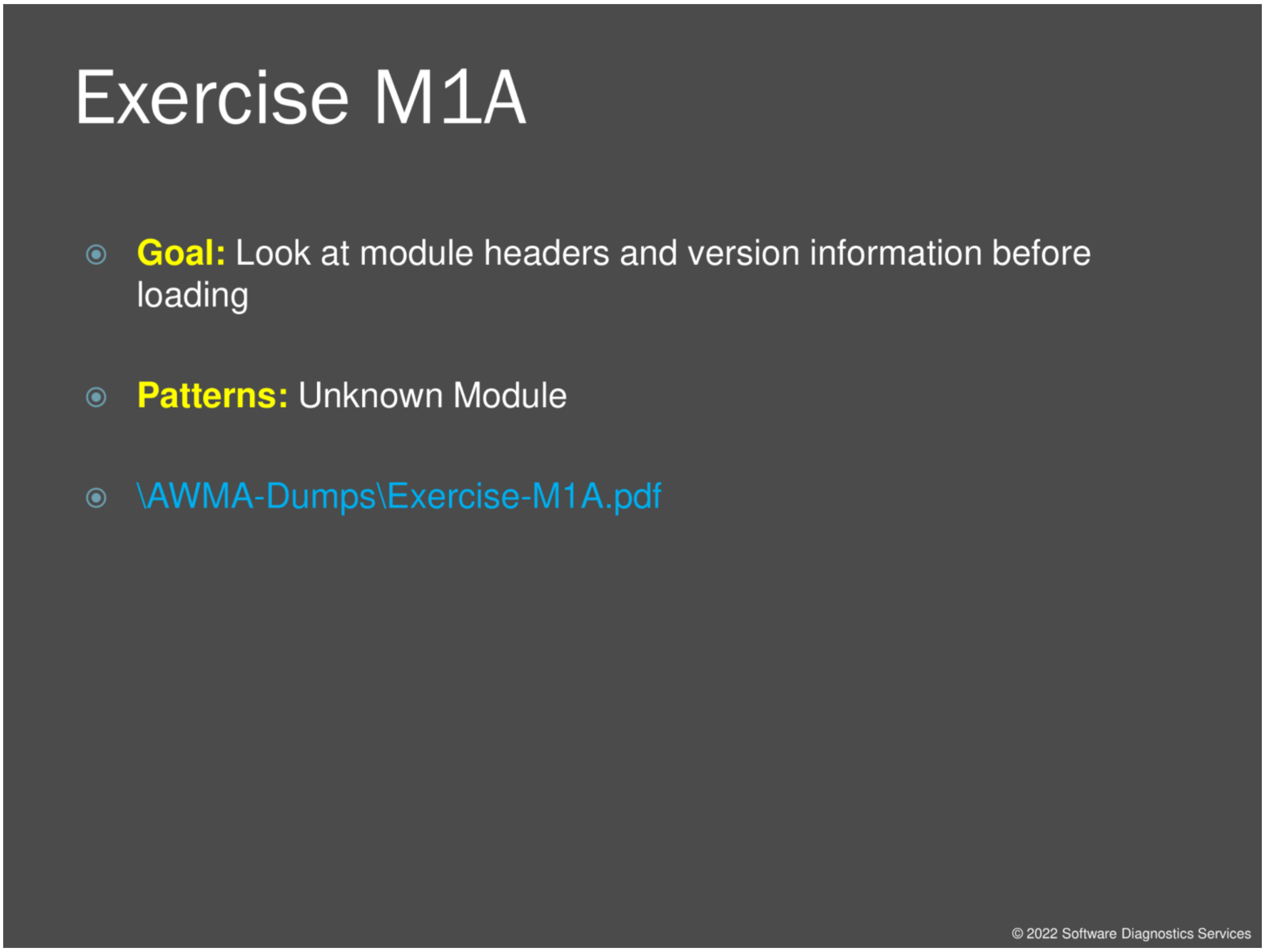
Here we provide a picture of a process space in 64-bit Windows. You see, user space is no longer restricted to 2 or 3 GB. On older x64 Windows systems, some DLLs are still loaded in the 2 GB address range as before, but many others are loaded at higher addresses. Newer systems, such as Windows 10 and 11, load all modules above 4GB range in virtual address space, as seen in the picture. We see that space distribution when we do a later exercise. But for now, we first look at executable files and DLLs.



EXE/DLL/SYS

```
0:000> lm
start             end                 module name
00007ffb`7ed20000 00007ffb`7ed3f000   M1DLL
[…]

0:000> dc 00007ffb`7ed20000  L40
00007ffb`7ed20000  00905a4d 00000003 00000004 0000ffff  MZ..............
[…]

0:000> !dh 00007ffb`7ed20000
[…]
```

Executable files, DLLs, and drivers (.SYS) all share the same format. The first comes an old MS-DOS header with an MZ signature and then a PE header or Portable Executable header that contains relative pointers or offsets to sections of code, data, and resources such as localized strings and dialog descriptions. However, anything can be stored as a resource, and we see that later in one of the exercises.



Exercise M1A

- **Goal:** Look at module headers and version information before loading
- **Patterns:** Unknown Module
- \AWMA-Dumps\Exercise-M1A.pdf

In addition to loading crash dumps in WinDbg, we can also load an executable or a DLL file as a crash dump. We do this in our first exercise.

Exercise M1A

Goal: Look at module headers and version information before loading.

Patterns: Unknown Module.

1. Launch WinDbg Preview.

2. Open \AWMA-Dumps\Executables\M1.exe.

3. We get the EXE file loaded:

```
Microsoft (R) Windows Debugger Version 10.0.25136.1001 AMD64
Copyright (c) Microsoft Corporation. All rights reserved.


Loading Dump File [C:\AWMA-Dumps\Executables\M1.exe]

************* Path validation summary **************
Response                         Time (ms)     Location
Deferred                                       srv*
Symbol search path is: srv*
Executable search path is:
ModLoad: 00000001`40000000 00000001`40020000   C:\AWMA-Dumps\Executables\M1.exe
*** WARNING: Unable to verify checksum for M1.exe
M1+0x1748:
00000001`40001748 4883ec28        sub     rsp,28h
```

4. Open a log file:

```
0:000> .logopen C:\AWMA-Dumps\M1A.log
Opened log file 'C:\AWMA-Dumps\M1A.log'
```

5. **lmv** command lists module information:

```
0:000> lmv
start             end                 module name
00000001`40000000 00000001`40020000   M1       C (no symbols)
    Loaded symbol image file: M1.exe
    Mapped memory image file: C:\AWMA-Dumps\Executables\M1.exe
    Image path: C:\AWMA-Dumps\Executables\M1.exe
    Image name: M1.exe
    Browse all global symbols  functions  data
    Timestamp:        Mon Jul  4 18:01:41 2022 (62C31CF5)
    CheckSum:         00000000
    ImageSize:        00020000
    Translations:     0000.04b0 0000.04e4 0409.04b0 0409.04e4
    Information from resource tables:
```

Note module default load address.

6. **!lmi** command gives a bit more information:

```
0:000> !lmi 00000001`40000000
Loaded Module Info: [00000001`40000000]
         Module: M1
   Base Address: 0000000140000000
     Image Name: M1.exe
   Machine Type: 34404 (X64)
     Time Stamp: 62c31cf5 Mon Jul  4 18:01:41 2022
           Size: 20000
       CheckSum: 0
Characteristics: 22
Debug Data Dirs: Type  Size     VA  Pointer
             CODEVIEW    37, 16de8,   155e8 RSDS - GUID: {4DA766FB-D5ED-4091-9599-9C8098BCC72D}
               Age: 1, Pdb: C:\AWMA3\M1\x64\Release\M1.pdb
           VC_FEATURE    14, 16e20,   15620 [Data not mapped]
                 POGO   31c, 16e34,   15634 [Data not mapped]
                ILTCG     0,     0,       0  [Debug data not mapped]
     Image Type: FILE     - Image read successfully from debugger.
                 M1.exe
    Symbol Type: NONE     - PDB not found from symbol server.
    Load Report: no symbols loaded
```

Note a reference to a PDB file. If this reference is left by a developer, it might give some clues, as we see in other exercises.

7. We dump the first kilobyte:

```
0:000> dc 00000001`40000000 L400
00000001`40000000  00905a4d 00000003 00000004 0000ffff  MZ..............
00000001`40000010  000000b8 00000000 00000040 00000000  ........@.......
00000001`40000020  00000000 00000000 00000000 00000000  ................
00000001`40000030  00000000 00000000 00000000 00000110  ................
00000001`40000040  0eba1f0e cd09b400 4c01b821 685421cd  ........!..L.!Th
00000001`40000050  70207369 72676f72 63206d61 6f6e6e61  is program canno
00000001`40000060  65622074 6e757220 206e6920 20534f44  t be run in DOS
00000001`40000070  65646f6d 0a0d0d2e 00000024 00000000  mode....$.......
00000001`40000080  e5d9de50 b6b7bf14 b6b7bf14 b6b7bf14  P...............
00000001`40000090  b7b4c75f b6b7bf11 b7b2c75f b6b7bf9f  _......._.......
00000001`400000a0  b7b3c75f b6b7bf1e b7b2c574 b6b7bf3c  _.......t...<...
00000001`400000b0  b7b3c574 b6b7bf04 b7b4c574 b6b7bf1d  t.......t.......
00000001`400000c0  b7b6c75f b6b7bf11 b6b6bf14 b6b7bf79  _...........y...
00000001`400000d0  b7bec570 b6b7bf16 b648c570 b6b7bf15  p.......p.H.....
00000001`400000e0  b620bf14 b6b7bf15 b7b5c570 b6b7bf15  .. .....p.......
00000001`400000f0  68636952 b6b7bf14 00000000 00000000  Rich............
00000001`40000100  00000000 00000000 00000000 00000000  ................
00000001`40000110  00004550 00078664 62c31cf5 00000000  PE..d......b....
00000001`40000120  00000000 002200f0 200e020b 0000d400  ......".... ....
00000001`40000130  0000f200 00000000 00001748 00001000  ........H.......
00000001`40000140  40000000 00000001 00001000 00000200  ...@............
00000001`40000150  00000006 00000000 00000006 00000000  ................
00000001`40000160  00020000 00000400 00000000 81600002  ..............`.
00000001`40000170  00100000 00000000 00001000 00000000  ................
00000001`40000180  00100000 00000000 00001000 00000000  ................
00000001`40000190  00000000 00000010 00000000 00000000  ................
00000001`400001a0  00017f0c 0000003c 0001d000 00001d78  ....<.......x...
00000001`400001b0  0001b000 00000f30 00000000 00000000  ....0...........
00000001`400001c0  0001f000 00000660 000169f0 00000070  ....`....i..p...
00000001`400001d0  00000000 00000000 00000000 00000000  ................
```

```
00000001`400001e0  00000000 00000000 000168b0 00000140  .........h..@...
00000001`400001f0  00000000 00000000 0000f000 000002e8  ................
00000001`40000200  00000000 00000000 00000000 00000000  ................
00000001`40000210  00000000 00000000 7865742e 00000074  .........text...
00000001`40000220  0000d230 00001000 0000d400 00000400  0...............
00000001`40000230  00000000 00000000 00000000 60000020  ............ ..`
00000001`40000240  6164722e 00006174 000098ac 0000f000  .rdata..........
00000001`40000250  00009a00 0000d800 00000000 00000000  ................
00000001`40000260  00000000 40000040 7461642e 00000061  ....@..@.data...
00000001`40000270  00001ec0 00019000 00000c00 00017200  .............r..
00000001`40000280  00000000 00000000 00000000 c0000040  ............@...
00000001`40000290  6164702e 00006174 00000f30 0001b000  .pdata..0.......
00000001`400002a0  00001000 00017e00 00000000 00000000  .....~..........
00000001`400002b0  00000000 40000040 4144525f 00004154  ....@..@_RDATA..
00000001`400002c0  0000015c 0001c000 00000200 00018e00  \...............
00000001`400002d0  00000000 00000000 00000000 40000040  ............@..@
00000001`400002e0  7273722e 00000063 00001d78 0001d000  .rsrc...x.......
00000001`400002f0  00001e00 00019000 00000000 00000000  ................
00000001`40000300  00000000 40000040 6c65722e 0000636f  ....@..@.reloc..
00000001`40000310  00000660 0001f000 00000800 0001ae00  `...............
00000001`40000320  00000000 00000000 00000000 42000040  ............@..B
00000001`40000330  00000000 00000000 00000000 00000000  ................
00000001`40000340  00000000 00000000 00000000 00000000  ................
00000001`40000350  00000000 00000000 00000000 00000000  ................
00000001`40000360  00000000 00000000 00000000 00000000  ................
00000001`40000370  00000000 00000000 00000000 00000000  ................
00000001`40000380  00000000 00000000 00000000 00000000  ................
00000001`40000390  00000000 00000000 00000000 00000000  ................
00000001`400003a0  00000000 00000000 00000000 00000000  ................
00000001`400003b0  00000000 00000000 00000000 00000000  ................
00000001`400003c0  00000000 00000000 00000000 00000000  ................
00000001`400003d0  00000000 00000000 00000000 00000000  ................
00000001`400003e0  00000000 00000000 00000000 00000000  ................
00000001`400003f0  00000000 00000000 00000000 00000000  ................
[...]
```

8. **!dh** command dumps PE header:

```
0:000> !dh 00000001`40000000


File Type: EXECUTABLE IMAGE
FILE HEADER VALUES
    8664 machine (X64)
       7 number of sections
62C31CF5 time date stamp Mon Jul  4 18:01:41 2022

       0 file pointer to symbol table
       0 number of symbols
      F0 size of optional header
      22 characteristics
            Executable
            App can handle >2gb addresses

OPTIONAL HEADER VALUES
     20B magic #
   14.32 linker version
    D400 size of code
    F200 size of initialized data
       0 size of uninitialized data
    1748 address of entry point
    1000 base of code
         ----- new -----
0000000140000000 image base
    1000 section alignment
     200 file alignment
       2 subsystem (Windows GUI)
    6.00 operating system version
    0.00 image version
    6.00 subsystem version
   20000 size of image
     400 size of headers
       0 checksum
0000000000100000 size of stack reserve
0000000000001000 size of stack commit
0000000000100000 size of heap reserve
0000000000001000 size of heap commit
    8160  DLL characteristics
            High entropy VA supported
            Dynamic base
            NX compatible
            Terminal server aware
       0 [       0] address [size] of Export Directory
   17F0C [      3C] address [size] of Import Directory
   1D000 [    1D78] address [size] of Resource Directory
   1B000 [     F30] address [size] of Exception Directory
       0 [       0] address [size] of Security Directory
   1F000 [     660] address [size] of Base Relocation Directory
   169F0 [      70] address [size] of Debug Directory
       0 [       0] address [size] of Description Directory
       0 [       0] address [size] of Special Directory
       0 [       0] address [size] of Thread Storage Directory
   168B0 [     140] address [size] of Load Configuration Directory
       0 [       0] address [size] of Bound Import Directory
    F000 [     2E8] address [size] of Import Address Table Directory
       0 [       0] address [size] of Delay Import Directory
       0 [       0] address [size] of COR20 Header Directory
       0 [       0] address [size] of Reserved Directory


SECTION HEADER #1
   .text name
    D230 virtual size
    1000 virtual address
    D400 size of raw data
     400 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
60000020 flags
         Code
         (no align specified)
         Execute Read

SECTION HEADER #2
  .rdata name
    98AC virtual size
    F000 virtual address
    9A00 size of raw data
    D800 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
40000040 flags
         Initialized Data
         (no align specified)
         Read Only


Debug Directories(4)
      Type       Size     Address  Pointer
      cv           37       16de8    155e8   Format: RSDS, guid, 1,
C:\AWMA3\M1\x64\Release\M1.pdb
      (   12)      14       16e20    15620
      (   13)     31c       16e34    15634
      (   14)       0           0        0

SECTION HEADER #3
   .data name
    1EC0 virtual size
   19000 virtual address
     C00 size of raw data
   17200 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
C0000040 flags
         Initialized Data
         (no align specified)
         Read Write

SECTION HEADER #4
  .pdata name
     F30 virtual size
   1B000 virtual address
    1000 size of raw data
   17E00 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
40000040 flags
         Initialized Data
         (no align specified)
         Read Only

SECTION HEADER #5
  _RDATA name
     15C virtual size
   1C000 virtual address
     200 size of raw data
   18E00 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
40000040 flags
```

```
         Initialized Data
         (no align specified)
         Read Only

SECTION HEADER #6
   .rsrc name
    1D78 virtual size
   1D000 virtual address
    1E00 size of raw data
   19000 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
40000040 flags
         Initialized Data
         (no align specified)
         Read Only

SECTION HEADER #7
  .reloc name
     660 virtual size
   1F000 virtual address
     800 size of raw data
   1AE00 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
42000040 flags
         Initialized Data
         Discardable
         (no align specified)
         Read Only
```

Note **Import Directory, Import Address Table Directory,** and code **.text** section.

9. Let's look at **Import Address Table Directory** before dynamic linking takes place:

```
0:000> dps 00000001`40000000+F000
00000001`4000f000  ????????`????????
00000001`4000f008  ????????`????????
00000001`4000f010  ????????`????????
00000001`4000f018  ????????`????????
00000001`4000f020  ????????`????????
00000001`4000f028  ????????`????????
00000001`4000f030  ????????`????????
00000001`4000f038  ????????`????????
00000001`4000f040  ????????`????????
00000001`4000f048  ????????`????????
00000001`4000f050  ????????`????????
00000001`4000f058  ????????`????????
00000001`4000f060  ????????`????????
00000001`4000f068  ????????`????????
00000001`4000f070  ????????`????????
00000001`4000f078  ????????`????????
```

We see it is inaccessible or not present. However, **Import Directory** is available, and we can dump its contents using the module image address, relative offset, and size (in bytes). It is an array of structures, each of 5 double words (4 bytes per double word). This is why we use the **dd** command and divide the size by 4:

```
0:000> dd 00000001`40000000+17F0C L3C/4
00000001`40017f0c  00017f48 00000000 00000000 00018240
00000001`40017f1c  0000f000 00018190 00000000 00000000
00000001`40017f2c  00018388 0000f248 00000000 00000000
00000001`40017f3c  00000000 00000000 00000000
```

The first double word in each structure is a relative offset to a relative offset to an array of names such as function names, and the fourth double word is a relative offset to an import DLL name:

```
0:000> da 00000001`40000000+00018240
00000001`40018240  "KERNEL32.dll"

0:000> da 00000001`40000000+00018388
00000001`40018388  "USER32.dll"
```

We now examine function names to be imported from *KERNEL32.dll*:

```
0:000> dp 00000001`40000000+00017f48
00000001`40017f48  00000000`00018230 00000000`0001889c
00000001`40017f58  00000000`0001888e 00000000`00018880
00000001`40017f68  00000000`0001886c 00000000`0001885a
00000001`40017f78  00000000`00018844 00000000`00018830
00000001`40017f88  00000000`00018822 00000000`00018816
00000001`40017f98  00000000`00018804 00000000`000187f4
00000001`40017fa8  00000000`000187ea 00000000`000187dc
00000001`40017fb8  00000000`000187ce 00000000`00018394

0:000> dc 00000001`40000000+00000000`00018230 L100
00000001`40018230  6f4c03e7 694c6461 72617262 00005779  ..LoadLibraryW..
00000001`40018240  4e52454b 32334c45 6c6c642e 02680000  KERNEL32.dll..h.
00000001`40018250  64616f4c 69727453 0057676e 6f4c0253  LoadStringW.S.Lo
00000001`40018260  63416461 656c6563 6f746172 00577372  adAcceleratorsW.
00000001`40018270  6547018b 73654d74 65676173 03b40057  ..GetMessageW...
00000001`40018280  6e617254 74616c73 63634165 72656c65  TranslateAcceler
00000001`40018290  726f7461 03b60057 6e617254 74616c73  atorW...Translat
00000001`400182a0  73654d65 65676173 00bd0000 70736944  eMessage....Disp
00000001`400182b0  68637461 7373654d 57656761 025b0000  atchMessageW..[.
00000001`400182c0  64616f4c 6e6f6349 02590057 64616f4c  LoadIconW.Y.Load
00000001`400182d0  73727543 0057726f 655202df 74736967  CursorW...Regist
00000001`400182e0  6c437265 45737361 00005778 72430076  erClassExW..v.Cr
00000001`400182f0  65746165 646e6957 7845776f 03960057  eateWindowExW...
00000001`40018300  776f6853 646e6957 0000776f 705503d0  ShowWindow....Up
00000001`40018310  65746164 646e6957 0000776f 694400ba  dateWindow....Di
00000001`40018320  676f6c61 50786f42 6d617261 00b50057  alogBoxParamW...
00000001`40018330  74736544 57796f72 6f646e69 00a70077  DestroyWindow...
00000001`40018340  57666544 6f646e69 6f725077 00005763  DefWindowProcW..
00000001`40018350  65420011 506e6967 746e6961 00f40000  ..BeginPaint....
00000001`40018360  50646e45 746e6961 02af0000 74736f50  EndPaint....Post
00000001`40018370  74697551 7373654d 00656761 6e4500f2  QuitMessage...En
00000001`40018380  61694464 00676f6c 52455355 642e3233  dDialog.USER32.d
00000001`40018390  00006c6c 745204f5 7061436c 65727574  ll....RtlCapture
00000001`400183a0  746e6f43 00747865 745204fd 6f6f4c6c  Context...RtlLoo
00000001`400183b0  4670756b 74636e75 456e6f69 7972746e  kupFunctionEntry
00000001`400183c0  05040000 566c7452 75747269 6e556c61  ....RtlVirtualUn
00000001`400183d0  646e6977 05e60000 61686e55 656c646e  wind....Unhandle
00000001`400183e0  63784564 69747065 69466e6f 7265746c  dExceptionFilter
00000001`400183f0  05a40000 55746553 6e61686e 64656c64  ....SetUnhandled
00000001`40018400  65637845 6f697470 6c69466e 00726574  ExceptionFilter.
00000001`40018410  65470232 72754374 746e6572 636f7250  2.GetCurrentProc
00000001`40018420  00737365 655405c4 6e696d72 50657461  ess...TerminateP
00000001`40018430  65636f72 00007373 734903a8 636f7250  rocess....IsProc
00000001`40018440  6f737365 61654672 65727574 73657250  essorFeaturePres
00000001`40018450  00746e65 75510470 50797265 6f667265  ent.p.QueryPerfo
00000001`40018460  6e616d72 6f436563 65746e75 02330072  rmanceCounter.3.
00000001`40018470  43746547 65727275 7250746e 7365636f  GetCurrentProces
00000001`40018480  00644973 65470237 72754374 746e6572  sId.7.GetCurrent
00000001`40018490  65726854 64496461 030a0000 53746547  ThreadId....GetS
00000001`400184a0  65747379 6d69546d 46734165 54656c69  ystemTimeAsFileT
00000001`400184b0  00656d69 6e49038a 61697469 657a696c  ime...Initialize
00000001`400184c0  73694c53 61654874 03a00064 65447349  SListHead...IsDe
00000001`400184d0  67677562 72507265 6e657365 02f10074  buggerPresent...
00000001`400184e0  53746547 74726174 6e497075 00576f66  GetStartupInfoW.
00000001`400184f0  65470295 646f4d74 48656c75 6c646e61  ..GetModuleHandl
00000001`40018500  00005765 74520503 776e556c 45646e69  eW....RtlUnwindE
00000001`40018510  027d0078 4c746547 45747361 726f7272  x.}.GetLastError
00000001`40018520  05640000 4c746553 45747361 726f7272  ..d.SetLastError
00000001`40018530  01490000 65746e45 69724372 61636974  ..I.EnterCritica
00000001`40018540  6365536c 6e6f6974 03e00000 7661654c  lSection....Leav
00000001`40018550  69724365 61636974 6365536c 6e6f6974  eCriticalSection
00000001`40018560  01230000 656c6544 72436574 63697469  ..#.DeleteCritic
00000001`40018570  65536c61 6f697463 0386006e 74696e49  alSection...Init
00000001`40018580  696c6169 7243657a 63697469 65536c61  ializeCriticalSe
00000001`40018590  6f697463 646e416e 6e697053 6e756f43  ctionAndSpinCoun
00000001`400185a0  05d60074 41736c54 636f6c6c 05d80000  t...TlsAlloc....
00000001`400185b0  47736c54 61567465 0065756c 6c5405d9  TlsGetValue...Tl
00000001`400185c0  74655373 756c6156 05d70065 46736c54  sSetValue...TlsF
00000001`400185d0  00656572 724601c5 694c6565 72617262  ree...FreeLibrar
00000001`400185e0  02cd0079 50746547 41636f72 65726464  y...GetProcAddre
00000001`400185f0  00007373 6f4c03e6 694c6461 72617262  ss....LoadLibrar
00000001`40018600  57784579 01450000 6f636e45 6f506564  yExW..E.EncodePo
00000001`40018610  65746e69 04870072 73696152 63784565  inter...RaiseExc
00000001`40018620  69747065 00006e6f 745204ff 5463506c  eption....RtlPcT
```

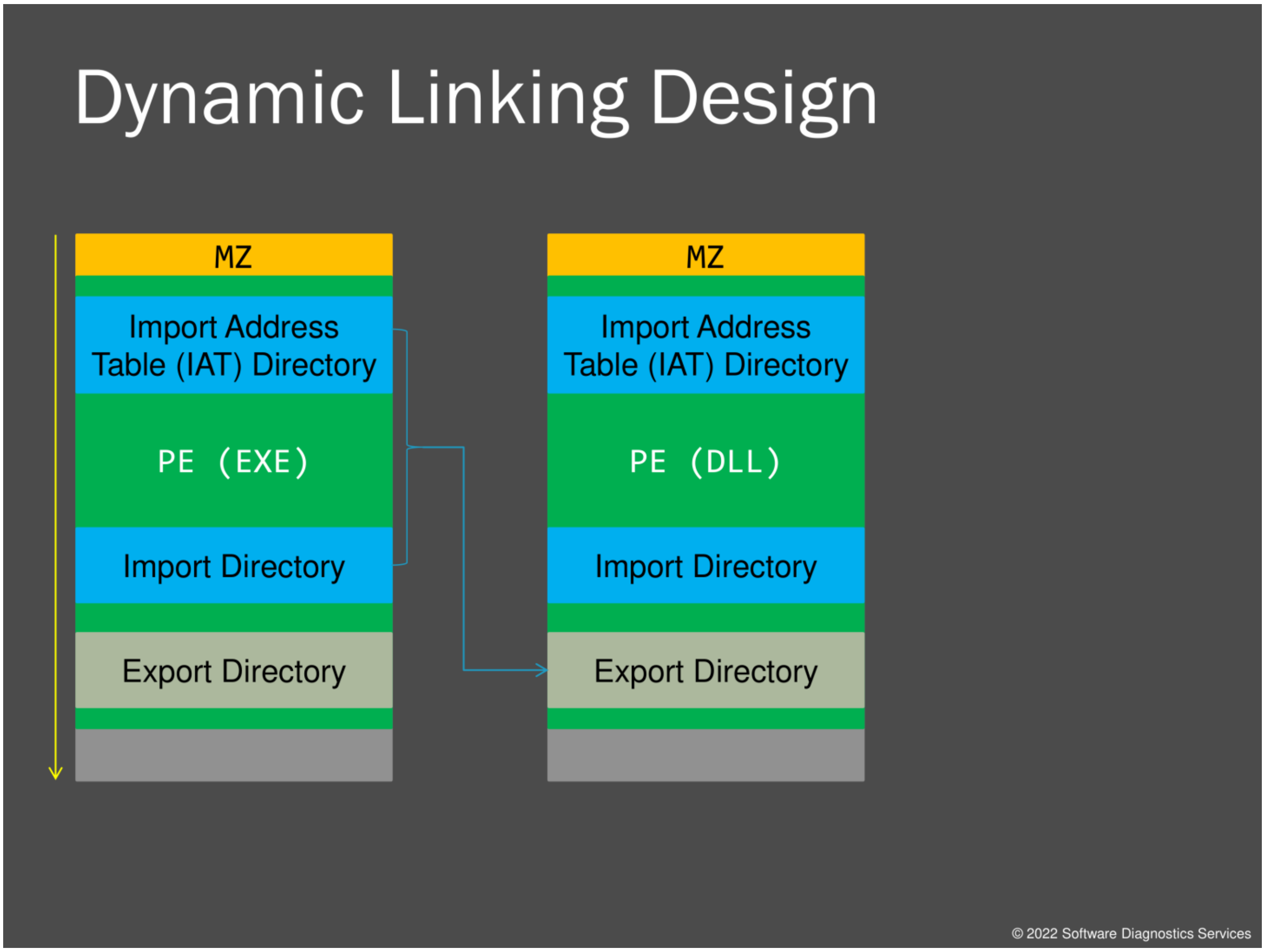
We can also get offsets by using **-i** or **-a** options for **!dh** command:

```
0:000> !dh -i 00000001`40000000
  _IMAGE_IMPORT_DESCRIPTOR 0000000140017f0c
    KERNEL32.dll
      000000014000F000 Import Address Table
      0000000140017F48 Import Name Table
                     0 time date stamp
                     0 Index of first forwarder reference


  _IMAGE_IMPORT_DESCRIPTOR 0000000140017f20
    USER32.dll
      000000014000F248 Import Address Table
      0000000140018190 Import Name Table
                     0 time date stamp
                     0 Index of first forwarder reference
```

10. Close the log file:

```
0:000> .logclose
Closing open log file C:\AWMA-Dumps\M1A.log
```

To avoid possible confusion and glitches, we recommend exiting WinDbg Preview or WinDbg after each exercise.



Dynamic Linking Design

MZ

Import Address Table (IAT) Directory

PE (EXE)

Import Directory

Export Directory

MZ

Import Address Table (IAT) Directory

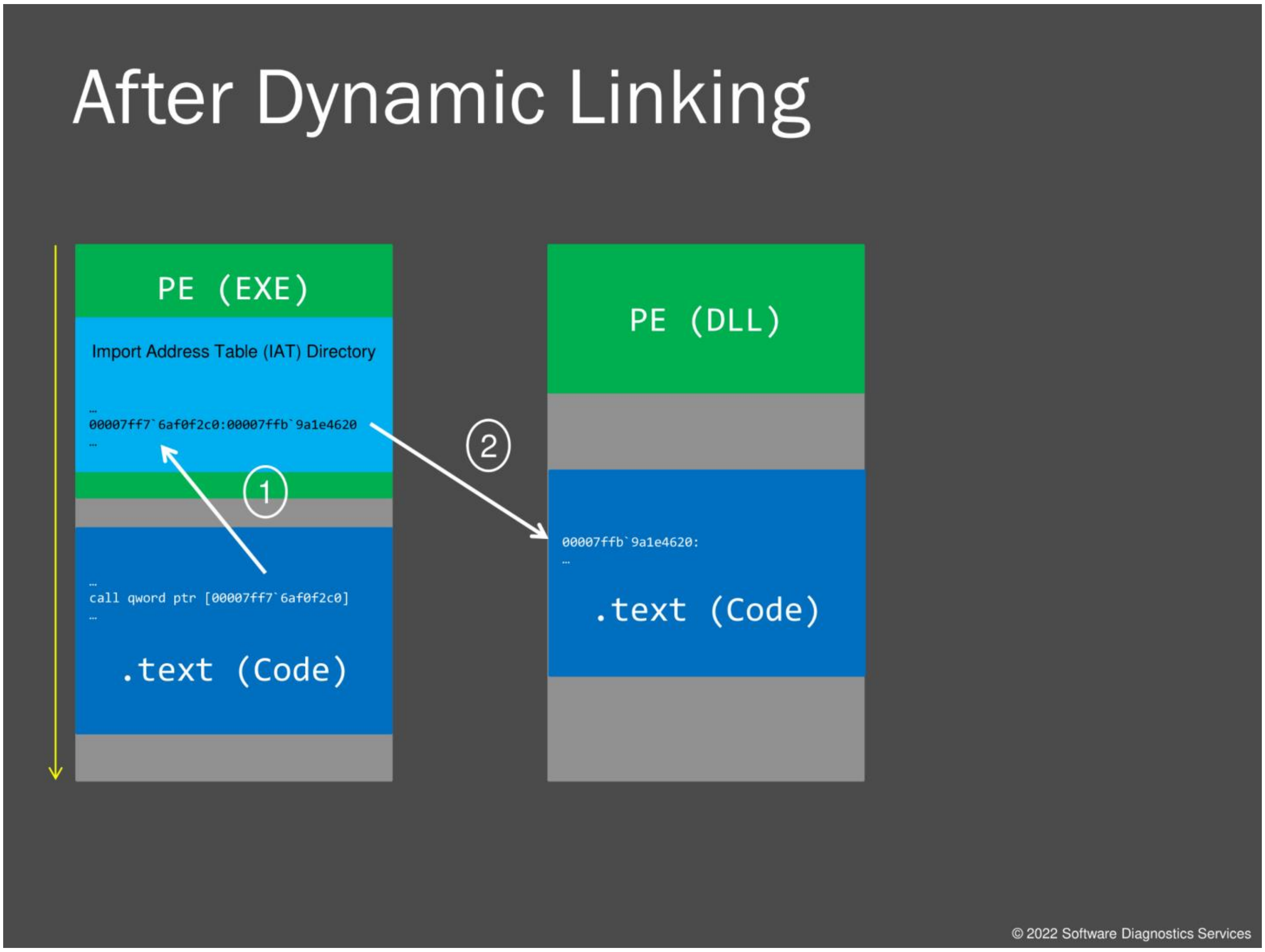
PE (DLL)

Import Directory

Export Directory

When a file such as an executable is loaded into memory, a runtime OS linker checks if that module references other DLL files. Recall that DLL means Dynamic Link Library. This is basically a collection of code and data that can be shared among processes. In a PE header, there is an Import Address Table that contains locations to store addresses of exported functions from another module. The same can also happen between DLLs; for example, user32.dll can reference ntdll.dll.

After Dynamic Linking



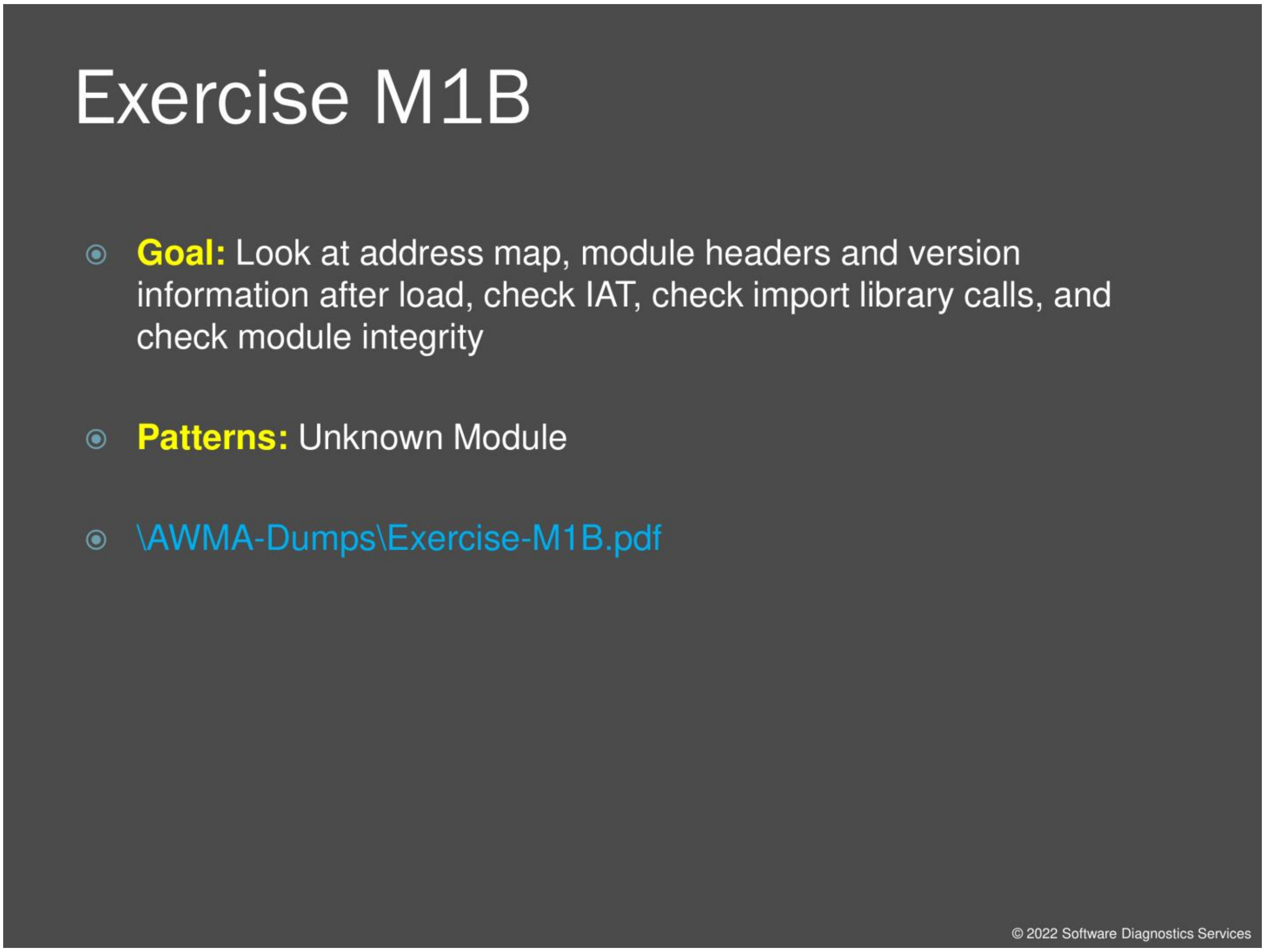
```
…
00007ff7`6af0f2c0:00007ffb`9a1e4620
…
```

```
…
call qword ptr [00007ff7`6af0f2c0]
…
```

```
00007ffb`9a1e4620:
…
```

© 2022 Software Diagnostics Services

Linking changes Import Address Table directory by substituting each entry with a real address from another already loaded DLL module. Code that transfers execution to such addresses uses indirect addressing. It uses an address from Import Address Table that points to code in another module. We see that during one of the exercises.



Exercise M1B

- **Goal:** Look at address map, module headers and version information after load, check IAT, check import library calls, and check module integrity
- **Patterns:** Unknown Module
- \AWMA-Dumps\Exercise-M1B.pdf

In our next exercise, we look at modules after dynamic linking had already been completed and process memory was saved.

Exercise M1B

Goal: Look at address map, module headers and version information after load, check IAT, check import library calls, and check module integrity.

Patterns: Unknown Module.

1. Launch WinDbg Preview.

2. Open \AWMA-Dumps\Processes\M1.dmp.

3. We get the dump file loaded:

```
Microsoft (R) Windows Debugger Version 10.0.25136.1001 AMD64
Copyright (c) Microsoft Corporation. All rights reserved.


Loading Dump File [C:\AWMA-Dumps\Processes\M1.dmp]
User Mini Dump File with Full Memory: Only application data is available


************* Path validation summary **************
Response                         Time (ms)     Location
Deferred                                       srv*
Symbol search path is: srv*
Executable search path is:
Windows 10 Version 22000 MP (8 procs) Free x64
Product: WinNt, suite: SingleUserTS
Edition build lab: 22000.1.amd64fre.co_release.210604-1628
Machine Name:
Debug session time: Mon Jul  4 18:02:40.000 2022 (UTC + 1:00)
System Uptime: 4 days 3:21:19.623
Process Uptime: 0 days 0:00:14.000
..............................
For analysis of this file, run !analyze -v
win32u!NtUserGetMessage+0x14:
00007ffb`994b1414 c3              ret
```

4. Open a log file:

```
0:000> .logopen C:\AWMA-Dumps\M1B.log
Opened log file 'C:\AWMA-Dumps\M1B.log'
```

5. **lmt** command lists modules and their timestamps:

```
0:000> lmt
start             end                 module name
00007ff7`6af00000 00007ff7`6af20000   M1       Mon Jul  4 18:01:41 2022 (62C31CF5)
00007ffb`76830000 00007ffb`76899000   oleacc    D4726D59 (This is a reproducible build file hash, not a timestamp)
00007ffb`7c810000 00007ffb`7c93d000   textinputframework  63938554 (This is a reproducible build file hash, not a timestamp)
00007ffb`7d9d0000 00007ffb`7da7e000   TextShaping   6627ED04 (This is a reproducible build file hash, not a timestamp)
00007ffb`7ed20000 00007ffb`7ed3f000   M1DLL    Mon Jul  4 18:01:38 2022 (62C31CF2)
00007ffb`8e9c0000 00007ffb`8ed2d000   CoreUIComponents   6685EB5C (This is a reproducible build file hash, not a timestamp)
00007ffb`91880000 00007ffb`919b2000   CoreMessaging  9E78ED02 (This is a reproducible build file hash, not a timestamp)
00007ffb`91c40000 00007ffb`91cd1000   apphelp   3C3AF44A (This is a reproducible build file hash, not a timestamp)
00007ffb`95100000 00007ffb`951ac000   uxtheme   E2C027FE (This is a reproducible build file hash, not a timestamp)
00007ffb`95cc0000 00007ffb`95e26000   WinTypes  B903EFEB (This is a reproducible build file hash, not a timestamp)
00007ffb`96e30000 00007ffb`96e48000   kernel_appcore  FB20135B (This is a reproducible build file hash, not a timestamp)
```

```
00007ffb`973f0000 00007ffb`973fc000   CRYPTBASE  14759998 (This is a reproducible build file hash, not a timestamp)
00007ffb`98c20000 00007ffb`98d32000   gdi32full  EB75EEDE (This is a reproducible build file hash, not a timestamp)
00007ffb`98e00000 00007ffb`98e9d000   msvcp_win  1FB7FD57 (This is a reproducible build file hash, not a timestamp)
00007ffb`98ea0000 00007ffb`99219000   KERNELBASE  89F799F7 (This is a reproducible build file hash, not a timestamp)
00007ffb`99390000 00007ffb`994a1000   ucrtbase  00E78CE9 (This is a reproducible build file hash, not a timestamp)
00007ffb`994b0000 00007ffb`994d6000   win32u    2EAB7211 (This is a reproducible build file hash, not a timestamp)
00007ffb`994e0000 00007ffb`9955f000   bcryptPrimitives  C0C2BD6F (This is a reproducible build file hash, not a timestamp)
00007ffb`99790000 00007ffb`9983e000   advapi32  69ED9A70 (This is a reproducible build file hash, not a timestamp)
00007ffb`99840000 00007ffb`99916000   oleaut32  F6E2D5CF (This is a reproducible build file hash, not a timestamp)
00007ffb`99920000 00007ffb`99c99000   combase   9E680117 (This is a reproducible build file hash, not a timestamp)
00007ffb`99ca0000 00007ffb`99d5d000   kernel32  AFEC8296 (This is a reproducible build file hash, not a timestamp)
00007ffb`99d60000 00007ffb`99e80000   rpcrt4    C1879A9E (This is a reproducible build file hash, not a timestamp)
00007ffb`9a1c0000 00007ffb`9a36c000   user32    95C2E8F0 (This is a reproducible build file hash, not a timestamp)
00007ffb`9ab30000 00007ffb`9abce000   sechost   62CBA37A (This is a reproducible build file hash, not a timestamp)
00007ffb`9abd0000 00007ffb`9ac73000   msvcrt    90483ED2 (This is a reproducible build file hash, not a timestamp)
00007ffb`9b290000 00007ffb`9b2b9000   gdi32     0B2998F3 (This is a reproducible build file hash, not a timestamp)
00007ffb`9b2c0000 00007ffb`9b3de000   msctf     53BBBDF3 (This is a reproducible build file hash, not a timestamp)
00007ffb`9b420000 00007ffb`9b451000   imm32     356942C7 (This is a reproducible build file hash, not a timestamp)
00007ffb`9b740000 00007ffb`9b949000   ntdll     B998B765 (This is a reproducible build file hash, not a timestamp)
```

Note the new module M1 load address. The latest Windows versions do not show real timestamps but reproducible build information for Microsoft modules. However, the presence of real timestamps may highlight the loaded 3rd-party modules.

6. Let's look at the address map (note how many regions are of a different type):

```
0:000> !address
```

```
Mapping file section regions...
Mapping module regions...
Mapping PEB regions...
Mapping TEB and stack regions...
Mapping heap regions...
Mapping page heap regions...
Mapping other regions...
Mapping stack trace database regions...
Mapping activation context regions...

        BaseAddress      EndAddress+1        RegionSize     Type       State                 Protect             Usage
--------------------------------------------------------------------------------------------------------------------------
+        0`00000000        0`7ffe0000        0`7ffe0000             MEM_FREE    PAGE_NOACCESS                      Free
+        0`7ffe0000        0`7ffe1000        0`00001000 MEM_PRIVATE MEM_COMMIT  PAGE_READONLY                      Other      [User Shared Data]
+        0`7ffe1000        0`7ffe6000        0`00005000             MEM_FREE    PAGE_NOACCESS                      Free
+        0`7ffe6000        0`7ffe7000        0`00001000 MEM_PRIVATE MEM_COMMIT  PAGE_READONLY                      <unknown>  [............M6.]
+        0`7ffe7000        7`ce460000        7`4e479000             MEM_FREE    PAGE_NOACCESS                      Free
+        7`ce460000        7`ce557000        0`000f7000 MEM_PRIVATE MEM_RESERVE                                    Stack      [~0; 6a08.8414]
         7`ce557000        7`ce55a000        0`00003000 MEM_PRIVATE MEM_COMMIT  PAGE_READWRITE | PAGE_GUARD        Stack      [~0; 6a08.8414]
         7`ce55a000        7`ce560000        0`00006000 MEM_PRIVATE MEM_COMMIT  PAGE_READWRITE                     Stack      [~0; 6a08.8414]
+        7`ce560000        7`ce600000        0`000a0000             MEM_FREE    PAGE_NOACCESS                      Free
+        7`ce600000        7`ce6fa000        0`000fa000 MEM_PRIVATE MEM_RESERVE                                    <unknown>
         7`ce6fa000        7`ce6fb000        0`00001000 MEM_PRIVATE MEM_COMMIT  PAGE_READWRITE                     PEB        [6a08]
         7`ce6fb000        7`ce6fd000        0`00002000 MEM_PRIVATE MEM_COMMIT  PAGE_READWRITE                     TEB        [~0; 6a08.8414]
         7`ce6fd000        7`ce6ff000        0`00002000 MEM_PRIVATE MEM_COMMIT  PAGE_READWRITE                     TEB        [~1; 6a08.46b8]
         7`ce6ff000        7`ce701000        0`00002000 MEM_PRIVATE MEM_COMMIT  PAGE_READWRITE                     TEB        [~2; 6a08.46b4]
         7`ce701000        7`ce703000        0`00002000 MEM_PRIVATE MEM_COMMIT  PAGE_READWRITE                     TEB        [~3; 6a08.1b78]
         7`ce703000        7`ce800000        0`000fd000 MEM_PRIVATE MEM_RESERVE                                    <unknown>
+        7`ce800000        7`ce8fb000        0`000fb000 MEM_PRIVATE MEM_RESERVE                                    Stack      [~1; 6a08.46b8]
         7`ce8fb000        7`ce8fe000        0`00003000 MEM_PRIVATE MEM_COMMIT  PAGE_READWRITE | PAGE_GUARD        Stack      [~1; 6a08.46b8]
         7`ce8fe000        7`ce900000        0`00002000 MEM_PRIVATE MEM_COMMIT  PAGE_READWRITE                     Stack      [~1; 6a08.46b8]
+        7`ce900000        7`ce9fb000        0`000fb000 MEM_PRIVATE MEM_RESERVE                                    Stack      [~2; 6a08.46b4]
         7`ce9fb000        7`ce9fe000        0`00003000 MEM_PRIVATE MEM_COMMIT  PAGE_READWRITE | PAGE_GUARD        Stack      [~2; 6a08.46b4]
         7`ce9fe000        7`cea00000        0`00002000 MEM_PRIVATE MEM_COMMIT  PAGE_READWRITE                     Stack      [~2; 6a08.46b4]
+        7`cea00000        7`ceafb000        0`000fb000 MEM_PRIVATE MEM_RESERVE                                    Stack      [~3; 6a08.1b78]
         7`ceafb000        7`ceafe000        0`00003000 MEM_PRIVATE MEM_COMMIT  PAGE_READWRITE | PAGE_GUARD        Stack      [~3; 6a08.1b78]
         7`ceafe000        7`ceb00000        0`00002000 MEM_PRIVATE MEM_COMMIT  PAGE_READWRITE                     Stack      [~3; 6a08.1b78]
+        7`ceb00000      180`2e440000      178`5f940000             MEM_FREE    PAGE_NOACCESS                      Free
+      180`2e440000      180`2e441000        0`00001000 MEM_MAPPED  MEM_COMMIT  PAGE_READONLY                      <unknown>  [........d.......]
+      180`2e441000      180`2e450000        0`0000f000             MEM_FREE    PAGE_NOACCESS                      Free
+      180`2e450000      180`2e451000        0`00001000 MEM_MAPPED  MEM_COMMIT  PAGE_READONLY                      <unknown>  [........d.......]
+      180`2e451000      180`2e460000        0`0000f000             MEM_FREE    PAGE_NOACCESS                      Free
+      180`2e460000      180`2e47f000        0`0001f000 MEM_MAPPED  MEM_COMMIT  PAGE_READONLY                      Other      [API Set Map]
+      180`2e47f000      180`2e480000        0`00001000             MEM_FREE    PAGE_NOACCESS                      Free
+      180`2e480000      180`2e484000        0`00004000 MEM_MAPPED  MEM_COMMIT  PAGE_READONLY                      Other      [System Default Activation Context Data]
+      180`2e484000      180`2e490000        0`0000c000             MEM_FREE    PAGE_NOACCESS                      Free
+      180`2e490000      180`2e491000        0`00001000 MEM_MAPPED  MEM_COMMIT  PAGE_READONLY                      Other      [Activation Context Data]
+      180`2e491000      180`2e4a0000        0`0000f000             MEM_FREE    PAGE_NOACCESS                      Free
+      180`2e4a0000      180`2e4a2000        0`00002000 MEM_PRIVATE MEM_COMMIT  PAGE_READWRITE                     <unknown>  [................]
+      180`2e4a2000      180`2e4b0000        0`0000e000             MEM_FREE    PAGE_NOACCESS                      Free
+      180`2e4b0000      180`2e4c1000        0`00011000 MEM_MAPPED  MEM_COMMIT  PAGE_READONLY                      <unknown>  [................]
+      180`2e4c1000      180`2e4d0000        0`0000f000             MEM_FREE    PAGE_NOACCESS                      Free
+      180`2e4d0000      180`2e4e1000        0`00011000 MEM_MAPPED  MEM_COMMIT  PAGE_READONLY                      <unknown>  [..R.............]
+      180`2e4e1000      180`2e4f0000        0`0000f000             MEM_FREE    PAGE_NOACCESS                      Free
+      180`2e4f0000      180`2e4f3000        0`00003000 MEM_MAPPED  MEM_COMMIT  PAGE_READONLY                      <unknown>  [........0...P...]
+      180`2e4f3000      180`2e500000        0`0000d000             MEM_FREE    PAGE_NOACCESS                      Free
+      180`2e500000      180`2e502000        0`00002000 MEM_PRIVATE MEM_COMMIT  PAGE_READWRITE                     Heap       [ID: 0; Handle: 000001802e6c0000; Type: Front End]
       180`2e502000      180`2e532000        0`00030000 MEM_PRIVATE MEM_RESERVE                                    Heap       [ID: 0; Handle: 000001802e6c0000; Type: Front End]
+      180`2e532000      180`2e540000        0`0000e000             MEM_FREE    PAGE_NOACCESS                      Free
+      180`2e540000      180`2e541000        0`00001000 MEM_MAPPED  MEM_COMMIT  PAGE_READONLY                      <unknown>  [............u...]
+      180`2e541000      180`2e550000        0`0000f000             MEM_FREE    PAGE_NOACCESS                      Free
+      180`2e550000      180`2e560000        0`00010000 MEM_MAPPED  MEM_COMMIT  PAGE_READWRITE                     Heap       [ID: 1; Handle: 000001802e550000; Type: Segment]
+      180`2e560000      180`2e563000        0`00003000 MEM_MAPPED  MEM_COMMIT  PAGE_READONLY                      <unknown>  [........0...P...]
+      180`2e563000      180`2e570000        0`0000d000             MEM_FREE    PAGE_NOACCESS                      Free
+      180`2e570000      180`2e63e000        0`000ce000 MEM_MAPPED  MEM_COMMIT  PAGE_READONLY                      <unknown>  [................]
+      180`2e63e000      180`2e640000        0`00002000             MEM_FREE    PAGE_NOACCESS                      Free
+      180`2e640000      180`2e651000        0`00011000 MEM_MAPPED  MEM_COMMIT  PAGE_READONLY                      <unknown>  [................]
+      180`2e651000      180`2e660000        0`0000f000             MEM_FREE    PAGE_NOACCESS                      Free
+      180`2e660000      180`2e671000        0`00011000 MEM_MAPPED  MEM_COMMIT  PAGE_READONLY                      <unknown>  [..R.............]
+      180`2e671000      180`2e680000        0`0000f000             MEM_FREE    PAGE_NOACCESS                      Free
+      180`2e680000      180`2e684000        0`00004000 MEM_MAPPED  MEM_COMMIT  PAGE_READONLY                      <unknown>  [........mEq..X..]
       180`2e684000      180`2e688000        0`00004000 MEM_MAPPED  MEM_RESERVE                                    <unknown>
+      180`2e688000      180`2e690000        0`00008000             MEM_FREE    PAGE_NOACCESS                      Free
+        180`2e690000      180`2e691000        0`00001000 MEM_PRIVATE MEM_COMMIT  PAGE_READWRITE                    <unknown>  [................]
+        180`2e691000      180`2e6a0000        0`0000f000             MEM_FREE    PAGE_NOACCESS                     Free
+        180`2e6a0000      180`2e6a1000        0`00001000 MEM_PRIVATE MEM_COMMIT  PAGE_READWRITE                    <unknown>  [................]
+        180`2e6a1000      180`2e6b0000        0`0000f000             MEM_FREE    PAGE_NOACCESS                     Free
+        180`2e6b0000      180`2e6b5000        0`00005000 MEM_MAPPED  MEM_COMMIT  PAGE_READONLY                     <unknown>  [................]
+        180`2e6b5000      180`2e6c0000        0`0000b000             MEM_FREE    PAGE_NOACCESS                     Free
+        180`2e6c0000      180`2e726000        0`00066000 MEM_PRIVATE MEM_COMMIT  PAGE_READWRITE                    Heap       [ID: 0; Handle: 000001802e6c0000; Type: Segment]
         180`2e726000      180`2e7bf000        0`00099000 MEM_PRIVATE MEM_RESERVE                                   Heap       [ID: 0; Handle: 000001802e6c0000; Type: Segment]
         180`2e7bf000      180`2e7c0000        0`00001000 MEM_PRIVATE MEM_RESERVE                                   <unknown>
+        180`2e7c0000      180`2e802000        0`00042000 MEM_MAPPED  MEM_COMMIT  PAGE_READONLY                     <unknown>  [................]
         180`2e802000      180`2e9c0000        0`001be000 MEM_MAPPED  MEM_RESERVE                                   <unknown>
+        180`2e9c0000      180`2eb41000        0`00181000 MEM_MAPPED  MEM_COMMIT  PAGE_READONLY                     Other      [GDI Shared Handle Table]
+        180`2eb41000      180`2eb50000        0`0000f000             MEM_FREE    PAGE_NOACCESS                     Free
+        180`2eb50000      180`2edb8000        0`00268000 MEM_MAPPED  MEM_COMMIT  PAGE_READONLY                     <unknown>  [.............X..]
         180`2edb8000      180`2ff51000        0`01199000 MEM_MAPPED  MEM_RESERVE                                   <unknown>
+        180`2ff51000      180`2ff60000        0`0000f000             MEM_FREE    PAGE_NOACCESS                     Free
+        180`2ff60000      180`2ff62000        0`00002000 MEM_PRIVATE MEM_COMMIT  PAGE_READWRITE                    Heap       [ID: 3; Handle: 0000018030000000; Type: Front End]
         180`2ff62000      180`2ff92000        0`00030000 MEM_PRIVATE MEM_RESERVE                                   Heap       [ID: 3; Handle: 0000018030000000; Type: Front End]
+        180`2ff92000      180`2ffa0000        0`0000e000             MEM_FREE    PAGE_NOACCESS                     Free
+        180`2ffa0000      180`2ffa1000        0`00001000 MEM_PRIVATE MEM_COMMIT  PAGE_READWRITE                    Heap       [ID: 2; Handle: 0000018030130000; Type: Front End]
         180`2ffa1000      180`2ffd2000        0`00031000 MEM_PRIVATE MEM_RESERVE                                   Heap       [ID: 2; Handle: 0000018030130000; Type: Front End]
+        180`2ffd2000      180`2ffe0000        0`0000e000             MEM_FREE    PAGE_NOACCESS                     Free
+        180`2ffe0000      180`2ffe1000        0`00001000 MEM_MAPPED  MEM_COMMIT  PAGE_READONLY                     Other      [Activation Context Data]
+        180`2ffe1000      180`2fff0000        0`0000f000             MEM_FREE    PAGE_NOACCESS                     Free
+        180`2fff0000      180`2fff3000        0`00003000 MEM_MAPPED  MEM_COMMIT  PAGE_READONLY                     <unknown>  [MZ..............]
+        180`2fff3000      180`30000000        0`0000d000             MEM_FREE    PAGE_NOACCESS                     Free
+        180`30000000      180`3000f000        0`0000f000 MEM_PRIVATE MEM_COMMIT  PAGE_READWRITE                    Heap       [ID: 3; Handle: 0000018030000000; Type: Segment]
         180`3000f000      180`30010000        0`00001000 MEM_PRIVATE MEM_RESERVE                                   <unknown>
+        180`30010000      180`30011000        0`00001000 MEM_PRIVATE MEM_COMMIT  PAGE_READWRITE                    <unknown>  [........P.p.....]
         180`30011000      180`30110000        0`000ff000 MEM_PRIVATE MEM_RESERVE                                   <unknown>
+        180`30110000      180`30130000        0`00020000             MEM_FREE    PAGE_NOACCESS                     Free
+        180`30130000      180`30137000        0`00007000 MEM_PRIVATE MEM_COMMIT  PAGE_READWRITE                    Heap       [ID: 2; Handle: 0000018030130000; Type: Segment]
         180`30137000      180`3013f000        0`00008000 MEM_PRIVATE MEM_RESERVE                                   Heap       [ID: 2; Handle: 0000018030130000; Type: Segment]
         180`3013f000      180`30140000        0`00001000 MEM_PRIVATE MEM_RESERVE                                   <unknown>
+        180`30140000      180`30265000        0`00125000 MEM_MAPPED  MEM_COMMIT  PAGE_READONLY                     <unknown>  [BEGINTHM.....Ein]
+        180`30265000      180`30270000        0`0000b000             MEM_FREE    PAGE_NOACCESS                     Free
+        180`30270000      180`305aa000        0`0033a000 MEM_MAPPED  MEM_COMMIT  PAGE_READONLY                     <unknown>  [........hl..x...]
+        180`305aa000      180`305b0000        0`00006000             MEM_FREE    PAGE_NOACCESS                     Free
+        180`305b0000      180`305b1000        0`00001000 MEM_PRIVATE MEM_COMMIT  PAGE_READWRITE                    <unknown>  [................]
         180`305b1000      180`30db0000        0`007ff000 MEM_PRIVATE MEM_RESERVE                                   <unknown>
+        180`30db0000      180`31960000        0`00bb0000             MEM_FREE    PAGE_NOACCESS                     Free
+        180`31960000      180`32b00000        0`011a0000 MEM_MAPPED  MEM_COMMIT  PAGE_READONLY                     <unknown>  [..W.............]
+        180`32b00000     7ff4`fd070000     7e74`ca570000             MEM_FREE    PAGE_NOACCESS                     Free
+       7ff4`fd070000     7ff4`fd075000        0`00005000 MEM_MAPPED  MEM_COMMIT  PAGE_READONLY                     Other      [Read Only Shared Memory]
        7ff4`fd075000     7ff4`fd170000        0`000fb000 MEM_MAPPED  MEM_RESERVE                                   <unknown>
+       7ff4`fd170000     7ff5`fd190000        1`00020000 MEM_PRIVATE MEM_RESERVE                                   <unknown>
+       7ff5`fd190000     7ff5`ff190000        0`02000000 MEM_PRIVATE MEM_RESERVE                                   <unknown>
        7ff5`ff190000     7ff5`ff191000        0`00001000 MEM_PRIVATE MEM_COMMIT  PAGE_READWRITE                    <unknown>  [................]
+       7ff5`ff191000     7ff5`ff1a0000        0`0000f000             MEM_FREE    PAGE_NOACCESS                     Free
+       7ff5`ff1a0000     7ff5`ff1a1000        0`00001000 MEM_MAPPED  MEM_COMMIT  PAGE_READONLY                     <unknown>  [................]
+       7ff5`ff1a1000     7ff7`6af00000        1`6bd5f000             MEM_FREE    PAGE_NOACCESS                     Free
+       7ff7`6af00000     7ff7`6af01000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                     Image      [M1; "C:\AWMA-Dumps\Executables\M1.exe"]
        7ff7`6af01000     7ff7`6af0f000        0`0000e000 MEM_IMAGE   MEM_COMMIT  PAGE_EXECUTE_READ                 Image      [M1; "C:\AWMA-Dumps\Executables\M1.exe"]
        7ff7`6af0f000     7ff7`6af19000        0`0000a000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                     Image      [M1; "C:\AWMA-Dumps\Executables\M1.exe"]
        7ff7`6af19000     7ff7`6af1b000        0`00002000 MEM_IMAGE   MEM_COMMIT  PAGE_READWRITE                    Image      [M1; "C:\AWMA-Dumps\Executables\M1.exe"]
        7ff7`6af1b000     7ff7`6af20000        0`00005000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                     Image      [M1; "C:\AWMA-Dumps\Executables\M1.exe"]
+       7ff7`6af20000     7ffb`76830000        4`0b910000             MEM_FREE    PAGE_NOACCESS                     Free
+       7ffb`76830000     7ffb`76831000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                     Image      [oleacc; "C:\Windows\System32\oleacc.dll"]
        7ffb`76831000     7ffb`76874000        0`00043000 MEM_IMAGE   MEM_COMMIT  PAGE_EXECUTE_READ                 Image      [oleacc; "C:\Windows\System32\oleacc.dll"]
        7ffb`76874000     7ffb`7688b000        0`00017000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                     Image      [oleacc; "C:\Windows\System32\oleacc.dll"]
        7ffb`7688b000     7ffb`7688c000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READWRITE                    Image      [oleacc; "C:\Windows\System32\oleacc.dll"]
        7ffb`7688c000     7ffb`76899000        0`0000d000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                     Image      [oleacc; "C:\Windows\System32\oleacc.dll"]
+       7ffb`76899000     7ffb`7c810000        0`05f77000             MEM_FREE    PAGE_NOACCESS                     Free
+       7ffb`7c810000     7ffb`7c811000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                     Image      [textinputframework;
"C:\Windows\System32\textinputframework.dll"]
        7ffb`7c811000     7ffb`7c8f3000        0`000e2000 MEM_IMAGE   MEM_COMMIT  PAGE_EXECUTE_READ                 Image      [textinputframework;
"C:\Windows\System32\textinputframework.dll"]
        7ffb`7c8f3000     7ffb`7c928000        0`00035000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                     Image      [textinputframework;
"C:\Windows\System32\textinputframework.dll"]
        7ffb`7c928000     7ffb`7c92b000        0`00003000 MEM_IMAGE   MEM_COMMIT  PAGE_READWRITE                    Image      [textinputframework;
"C:\Windows\System32\textinputframework.dll"]
        7ffb`7c92b000     7ffb`7c93d000        0`00012000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                     Image      [textinputframework;
"C:\Windows\System32\textinputframework.dll"]
+       7ffb`7c93d000     7ffb`7d9d0000        0`01093000             MEM_FREE    PAGE_NOACCESS                     Free
+       7ffb`7d9d0000     7ffb`7d9d1000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                     Image      [TextShaping; "C:\Windows\System32\TextShaping.dll"]
        7ffb`7d9d1000     7ffb`7da1d000        0`0004c000 MEM_IMAGE   MEM_COMMIT  PAGE_EXECUTE_READ                 Image      [TextShaping; "C:\Windows\System32\TextShaping.dll"]
        7ffb`7da1d000     7ffb`7da79000        0`0005c000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                     Image      [TextShaping; "C:\Windows\System32\TextShaping.dll"]
        7ffb`7da79000     7ffb`7da7a000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READWRITE                    Image      [TextShaping; "C:\Windows\System32\TextShaping.dll"]
        7ffb`7da7a000     7ffb`7da7e000        0`00004000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                     Image      [TextShaping; "C:\Windows\System32\TextShaping.dll"]
+       7ffb`7da7e000     7ffb`7ed20000        0`012a2000             MEM_FREE    PAGE_NOACCESS                     Free
+       7ffb`7ed20000     7ffb`7ed21000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                     Image      [M1DLL; "C:\AWMA-Dumps\Executables\M1DLL.dll"]
        7ffb`7ed21000     7ffb`7ed2f000        0`0000e000 MEM_IMAGE   MEM_COMMIT  PAGE_EXECUTE_READ                 Image      [M1DLL; "C:\AWMA-Dumps\Executables\M1DLL.dll"]
        7ffb`7ed2f000     7ffb`7ed39000        0`0000a000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                     Image      [M1DLL; "C:\AWMA-Dumps\Executables\M1DLL.dll"]
        7ffb`7ed39000     7ffb`7ed3b000        0`00002000 MEM_IMAGE   MEM_COMMIT  PAGE_READWRITE                    Image      [M1DLL; "C:\AWMA-Dumps\Executables\M1DLL.dll"]
        7ffb`7ed3b000     7ffb`7ed3f000        0`00004000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                     Image      [M1DLL; "C:\AWMA-Dumps\Executables\M1DLL.dll"]
+       7ffb`7ed3f000     7ffb`8e9c0000        0`0fc81000             MEM_FREE    PAGE_NOACCESS                     Free
+       7ffb`8e9c0000     7ffb`8e9c1000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                     Image      [CoreUIComponents; "C:\Windows\System32\CoreUIComponents.dll"]
        7ffb`8e9c1000     7ffb`8ebb6000        0`001f5000 MEM_IMAGE   MEM_COMMIT  PAGE_EXECUTE_READ                 Image      [CoreUIComponents; "C:\Windows\System32\CoreUIComponents.dll"]
        7ffb`8ebb6000     7ffb`8ecae000        0`000f8000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                     Image      [CoreUIComponents; "C:\Windows\System32\CoreUIComponents.dll"]
        7ffb`8ecae000     7ffb`8ecaf000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READWRITE                    Image      [CoreUIComponents; "C:\Windows\System32\CoreUIComponents.dll"]
        7ffb`8ecaf000     7ffb`8ecb0000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_WRITECOPY                    Image      [CoreUIComponents; "C:\Windows\System32\CoreUIComponents.dll"]
        7ffb`8ecb0000     7ffb`8ecb2000        0`00002000 MEM_IMAGE   MEM_COMMIT  PAGE_READWRITE                    Image      [CoreUIComponents; "C:\Windows\System32\CoreUIComponents.dll"]
        7ffb`8ecb2000     7ffb`8ed2d000        0`0007b000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                     Image      [CoreUIComponents; "C:\Windows\System32\CoreUIComponents.dll"]
+       7ffb`8ed2d000     7ffb`91880000        0`02b53000             MEM_FREE    PAGE_NOACCESS                     Free
+       7ffb`91880000     7ffb`91881000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                     Image      [CoreMessaging; "C:\Windows\System32\CoreMessaging.dll"]
        7ffb`91881000     7ffb`91953000        0`000d2000 MEM_IMAGE   MEM_COMMIT  PAGE_EXECUTE_READ                 Image      [CoreMessaging; "C:\Windows\System32\CoreMessaging.dll"]
        7ffb`91953000     7ffb`91990000        0`0003d000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                     Image      [CoreMessaging; "C:\Windows\System32\CoreMessaging.dll"]
        7ffb`91990000     7ffb`91992000        0`00002000 MEM_IMAGE   MEM_COMMIT  PAGE_READWRITE                    Image      [CoreMessaging; "C:\Windows\System32\CoreMessaging.dll"]
        7ffb`91992000     7ffb`919b2000        0`00020000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                     Image      [CoreMessaging; "C:\Windows\System32\CoreMessaging.dll"]
+       7ffb`919b2000     7ffb`91c40000        0`0028e000             MEM_FREE    PAGE_NOACCESS                     Free
+       7ffb`91c40000     7ffb`91c41000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                     Image      [apphelp; "C:\Windows\System32\apphelp.dll"]
        7ffb`91c41000     7ffb`91c8f000        0`0004e000 MEM_IMAGE   MEM_COMMIT  PAGE_EXECUTE_READ                 Image      [apphelp; "C:\Windows\System32\apphelp.dll"]
        7ffb`91c8f000     7ffb`91cb1000        0`00022000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                     Image      [apphelp; "C:\Windows\System32\apphelp.dll"]
        7ffb`91cb1000     7ffb`91cb4000        0`00003000 MEM_IMAGE   MEM_COMMIT  PAGE_READWRITE                    Image      [apphelp; "C:\Windows\System32\apphelp.dll"]
        7ffb`91cb4000     7ffb`91cd1000        0`0001d000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                     Image      [apphelp; "C:\Windows\System32\apphelp.dll"]
+       7ffb`91cd1000     7ffb`95100000        0`0342f000             MEM_FREE    PAGE_NOACCESS                     Free
+       7ffb`95100000     7ffb`95101000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                     Image      [uxtheme; "C:\Windows\System32\uxtheme.dll"]
        7ffb`95101000     7ffb`95169000        0`00068000 MEM_IMAGE   MEM_COMMIT  PAGE_EXECUTE_READ                 Image      [uxtheme; "C:\Windows\System32\uxtheme.dll"]
        7ffb`95169000     7ffb`9519e000        0`00035000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                     Image      [uxtheme; "C:\Windows\System32\uxtheme.dll"]
        7ffb`9519e000     7ffb`951a0000        0`00002000 MEM_IMAGE   MEM_COMMIT  PAGE_READWRITE                    Image      [uxtheme; "C:\Windows\System32\uxtheme.dll"]
        7ffb`951a0000     7ffb`951a1000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_WRITECOPY                    Image      [uxtheme; "C:\Windows\System32\uxtheme.dll"]
        7ffb`951a1000     7ffb`951ac000        0`0000b000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                     Image      [uxtheme; "C:\Windows\System32\uxtheme.dll"]
+       7ffb`951ac000     7ffb`95cc0000        0`00b14000             MEM_FREE    PAGE_NOACCESS                     Free
+       7ffb`95cc0000     7ffb`95cc1000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                     Image      [WinTypes; "C:\Windows\System32\WinTypes.dll"]
        7ffb`95cc1000     7ffb`95d45000        0`00084000 MEM_IMAGE   MEM_COMMIT  PAGE_EXECUTE_READ                 Image      [WinTypes; "C:\Windows\System32\WinTypes.dll"]
        7ffb`95d45000     7ffb`95e00000        0`000bb000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                     Image      [WinTypes; "C:\Windows\System32\WinTypes.dll"]
        7ffb`95e00000     7ffb`95e02000        0`00002000 MEM_IMAGE   MEM_COMMIT  PAGE_READWRITE                    Image      [WinTypes; "C:\Windows\System32\WinTypes.dll"]
        7ffb`95e02000     7ffb`95e26000        0`00024000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                     Image      [WinTypes; "C:\Windows\System32\WinTypes.dll"]
+       7ffb`95e26000     7ffb`96e30000        0`0100a000             MEM_FREE    PAGE_NOACCESS                     Free
+       7ffb`96e30000     7ffb`96e31000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                     Image      [kernel_appcore; "C:\Windows\System32\kernel.appcore.dll"]
        7ffb`96e31000     7ffb`96e3a000        0`00009000 MEM_IMAGE   MEM_COMMIT  PAGE_EXECUTE_READ                 Image      [kernel_appcore; "C:\Windows\System32\kernel.appcore.dll"]
```

```
       7ffb`96e3a000     7ffb`96e43000        0`00009000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [kernel_appcore; "C:\Windows\System32\kernel.appcore.dll"]
       7ffb`96e43000     7ffb`96e44000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READWRITE                     Image      [kernel_appcore; "C:\Windows\System32\kernel.appcore.dll"]
       7ffb`96e44000     7ffb`96e48000        0`00004000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [kernel_appcore; "C:\Windows\System32\kernel.appcore.dll"]
+      7ffb`96e48000     7ffb`973f0000        0`005a8000             MEM_FREE    PAGE_NOACCESS                      Free
+      7ffb`973f0000     7ffb`973f1000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [CRYPTBASE; "C:\Windows\System32\CRYPTBASE.DLL"]
       7ffb`973f1000     7ffb`973f4000        0`00003000 MEM_IMAGE   MEM_COMMIT  PAGE_EXECUTE_READ                  Image      [CRYPTBASE; "C:\Windows\System32\CRYPTBASE.DLL"]
       7ffb`973f4000     7ffb`973f7000        0`00003000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [CRYPTBASE; "C:\Windows\System32\CRYPTBASE.DLL"]
       7ffb`973f7000     7ffb`973f8000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READWRITE                     Image      [CRYPTBASE; "C:\Windows\System32\CRYPTBASE.DLL"]
       7ffb`973f8000     7ffb`973fc000        0`00004000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [CRYPTBASE; "C:\Windows\System32\CRYPTBASE.DLL"]
+      7ffb`973fc000     7ffb`98c20000        0`01824000             MEM_FREE    PAGE_NOACCESS                      Free
+      7ffb`98c20000     7ffb`98c21000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [gdi32full; "C:\Windows\System32\gdi32full.dll"]
       7ffb`98c21000     7ffb`98cc2000        0`000a1000 MEM_IMAGE   MEM_COMMIT  PAGE_EXECUTE_READ                  Image      [gdi32full; "C:\Windows\System32\gdi32full.dll"]
       7ffb`98cc2000     7ffb`98d11000        0`0004f000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [gdi32full; "C:\Windows\System32\gdi32full.dll"]
       7ffb`98d11000     7ffb`98d16000        0`00005000 MEM_IMAGE   MEM_COMMIT  PAGE_READWRITE                     Image      [gdi32full; "C:\Windows\System32\gdi32full.dll"]
       7ffb`98d16000     7ffb`98d32000        0`0001c000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [gdi32full; "C:\Windows\System32\gdi32full.dll"]
+      7ffb`98d32000     7ffb`98e00000        0`000ce000             MEM_FREE    PAGE_NOACCESS                      Free
+      7ffb`98e00000     7ffb`98e01000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [msvcp_win; "C:\Windows\System32\msvcp_win.dll"]
       7ffb`98e01000     7ffb`98e56000        0`00055000 MEM_IMAGE   MEM_COMMIT  PAGE_EXECUTE_READ                  Image      [msvcp_win; "C:\Windows\System32\msvcp_win.dll"]
       7ffb`98e56000     7ffb`98e91000        0`0003b000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [msvcp_win; "C:\Windows\System32\msvcp_win.dll"]
       7ffb`98e91000     7ffb`98e92000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_WRITECOPY                     Image      [msvcp_win; "C:\Windows\System32\msvcp_win.dll"]
       7ffb`98e92000     7ffb`98e95000        0`00003000 MEM_IMAGE   MEM_COMMIT  PAGE_READWRITE                     Image      [msvcp_win; "C:\Windows\System32\msvcp_win.dll"]
       7ffb`98e95000     7ffb`98e9d000        0`00008000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [msvcp_win; "C:\Windows\System32\msvcp_win.dll"]
+      7ffb`98e9d000     7ffb`98ea0000        0`00003000             MEM_FREE    PAGE_NOACCESS                      Free
+      7ffb`98ea0000     7ffb`98ea1000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [KERNELBASE; "C:\Windows\System32\KERNELBASE.dll"]
       7ffb`98ea1000     7ffb`9901a000        0`00179000 MEM_IMAGE   MEM_COMMIT  PAGE_EXECUTE_READ                  Image      [KERNELBASE; "C:\Windows\System32\KERNELBASE.dll"]
       7ffb`9901a000     7ffb`991cc000        0`001b2000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [KERNELBASE; "C:\Windows\System32\KERNELBASE.dll"]
       7ffb`991cc000     7ffb`991d1000        0`00005000 MEM_IMAGE   MEM_COMMIT  PAGE_READWRITE                     Image      [KERNELBASE; "C:\Windows\System32\KERNELBASE.dll"]
       7ffb`991d1000     7ffb`99219000        0`00048000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [KERNELBASE; "C:\Windows\System32\KERNELBASE.dll"]
+      7ffb`99219000     7ffb`99390000        0`00177000             MEM_FREE    PAGE_NOACCESS                      Free
+      7ffb`99390000     7ffb`99391000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [ucrtbase; "C:\Windows\System32\ucrtbase.dll"]
       7ffb`99391000     7ffb`99454000        0`000c3000 MEM_IMAGE   MEM_COMMIT  PAGE_EXECUTE_READ                  Image      [ucrtbase; "C:\Windows\System32\ucrtbase.dll"]
       7ffb`99454000     7ffb`9948f000        0`0003b000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [ucrtbase; "C:\Windows\System32\ucrtbase.dll"]
       7ffb`9948f000     7ffb`99492000        0`00003000 MEM_IMAGE   MEM_COMMIT  PAGE_READWRITE                     Image      [ucrtbase; "C:\Windows\System32\ucrtbase.dll"]
       7ffb`99492000     7ffb`994a1000        0`0000f000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [ucrtbase; "C:\Windows\System32\ucrtbase.dll"]
+      7ffb`994a1000     7ffb`994b0000        0`0000f000             MEM_FREE    PAGE_NOACCESS                      Free
+      7ffb`994b0000     7ffb`994b1000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [win32u; "C:\Windows\System32\win32u.dll"]
       7ffb`994b1000     7ffb`994bd000        0`0000c000 MEM_IMAGE   MEM_COMMIT  PAGE_EXECUTE_READ                  Image      [win32u; "C:\Windows\System32\win32u.dll"]
       7ffb`994bd000     7ffb`994ce000        0`00011000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [win32u; "C:\Windows\System32\win32u.dll"]
       7ffb`994ce000     7ffb`994cf000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READWRITE                     Image      [win32u; "C:\Windows\System32\win32u.dll"]
       7ffb`994cf000     7ffb`994d6000        0`00007000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [win32u; "C:\Windows\System32\win32u.dll"]
+      7ffb`994d6000     7ffb`994e0000        0`0000a000             MEM_FREE    PAGE_NOACCESS                      Free
+      7ffb`994e0000     7ffb`994e1000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [bcryptPrimitives; "C:\Windows\System32\bcryptPrimitives.dll"]
       7ffb`994e1000     7ffb`99542000        0`00061000 MEM_IMAGE   MEM_COMMIT  PAGE_EXECUTE_READ                  Image      [bcryptPrimitives; "C:\Windows\System32\bcryptPrimitives.dll"]
       7ffb`99542000     7ffb`99558000        0`00016000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [bcryptPrimitives; "C:\Windows\System32\bcryptPrimitives.dll"]
       7ffb`99558000     7ffb`99559000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READWRITE                     Image      [bcryptPrimitives; "C:\Windows\System32\bcryptPrimitives.dll"]
       7ffb`99559000     7ffb`9955f000        0`00006000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [bcryptPrimitives; "C:\Windows\System32\bcryptPrimitives.dll"]
+      7ffb`9955f000     7ffb`99790000        0`00231000             MEM_FREE    PAGE_NOACCESS                      Free
+      7ffb`99790000     7ffb`99791000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [advapi32; "C:\Windows\System32\advapi32.dll"]
       7ffb`99791000     7ffb`997f9000        0`00068000 MEM_IMAGE   MEM_COMMIT  PAGE_EXECUTE_READ                  Image      [advapi32; "C:\Windows\System32\advapi32.dll"]
       7ffb`997f9000     7ffb`99830000        0`00037000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [advapi32; "C:\Windows\System32\advapi32.dll"]
       7ffb`99830000     7ffb`99831000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READWRITE                     Image      [advapi32; "C:\Windows\System32\advapi32.dll"]
       7ffb`99831000     7ffb`99832000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_WRITECOPY                     Image      [advapi32; "C:\Windows\System32\advapi32.dll"]
       7ffb`99832000     7ffb`99834000        0`00002000 MEM_IMAGE   MEM_COMMIT  PAGE_READWRITE                     Image      [advapi32; "C:\Windows\System32\advapi32.dll"]
       7ffb`99834000     7ffb`99835000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_WRITECOPY                     Image      [advapi32; "C:\Windows\System32\advapi32.dll"]
       7ffb`99835000     7ffb`9983e000        0`00009000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [advapi32; "C:\Windows\System32\advapi32.dll"]
+      7ffb`9983e000     7ffb`99840000        0`00002000             MEM_FREE    PAGE_NOACCESS                      Free
+      7ffb`99840000     7ffb`99841000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [oleaut32; "C:\Windows\System32\oleaut32.dll"]
       7ffb`99841000     7ffb`998de000        0`0009d000 MEM_IMAGE   MEM_COMMIT  PAGE_EXECUTE_READ                  Image      [oleaut32; "C:\Windows\System32\oleaut32.dll"]
       7ffb`998de000     7ffb`99904000        0`00026000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [oleaut32; "C:\Windows\System32\oleaut32.dll"]
       7ffb`99904000     7ffb`99907000        0`00003000 MEM_IMAGE   MEM_COMMIT  PAGE_READWRITE                     Image      [oleaut32; "C:\Windows\System32\oleaut32.dll"]
       7ffb`99907000     7ffb`99916000        0`0000f000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [oleaut32; "C:\Windows\System32\oleaut32.dll"]
+      7ffb`99916000     7ffb`99920000        0`0000a000             MEM_FREE    PAGE_NOACCESS                      Free
+      7ffb`99920000     7ffb`99921000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [combase; "C:\Windows\System32\combase.dll"]
       7ffb`99921000     7ffb`99b82000        0`00261000 MEM_IMAGE   MEM_COMMIT  PAGE_EXECUTE_READ                  Image      [combase; "C:\Windows\System32\combase.dll"]
       7ffb`99b82000     7ffb`99c44000        0`000c2000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [combase; "C:\Windows\System32\combase.dll"]
       7ffb`99c44000     7ffb`99c4a000        0`00006000 MEM_IMAGE   MEM_COMMIT  PAGE_READWRITE                     Image      [combase; "C:\Windows\System32\combase.dll"]
       7ffb`99c4a000     7ffb`99c99000        0`0004f000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [combase; "C:\Windows\System32\combase.dll"]
+      7ffb`99c99000     7ffb`99ca0000        0`00007000             MEM_FREE    PAGE_NOACCESS                      Free
+      7ffb`99ca0000     7ffb`99ca1000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [kernel32; "C:\Windows\System32\kernel32.dll"]
       7ffb`99ca1000     7ffb`99d1e000        0`0007d000 MEM_IMAGE   MEM_COMMIT  PAGE_EXECUTE_READ                  Image      [kernel32; "C:\Windows\System32\kernel32.dll"]
       7ffb`99d1e000     7ffb`99d52000        0`00034000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [kernel32; "C:\Windows\System32\kernel32.dll"]
       7ffb`99d52000     7ffb`99d53000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READWRITE                     Image      [kernel32; "C:\Windows\System32\kernel32.dll"]
       7ffb`99d53000     7ffb`99d54000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_WRITECOPY                     Image      [kernel32; "C:\Windows\System32\kernel32.dll"]
       7ffb`99d54000     7ffb`99d5d000        0`00009000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [kernel32; "C:\Windows\System32\kernel32.dll"]
+      7ffb`99d5d000     7ffb`99d60000        0`00003000             MEM_FREE    PAGE_NOACCESS                      Free
+      7ffb`99d60000     7ffb`99d61000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [rpcrt4; "C:\Windows\System32\rpcrt4.dll"]
       7ffb`99d61000     7ffb`99e40000        0`000df000 MEM_IMAGE   MEM_COMMIT  PAGE_EXECUTE_READ                  Image      [rpcrt4; "C:\Windows\System32\rpcrt4.dll"]
       7ffb`99e40000     7ffb`99e69000        0`00029000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [rpcrt4; "C:\Windows\System32\rpcrt4.dll"]
       7ffb`99e69000     7ffb`99e6b000        0`00002000 MEM_IMAGE   MEM_COMMIT  PAGE_READWRITE                     Image      [rpcrt4; "C:\Windows\System32\rpcrt4.dll"]
       7ffb`99e6b000     7ffb`99e80000        0`00015000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [rpcrt4; "C:\Windows\System32\rpcrt4.dll"]
+      7ffb`99e80000     7ffb`9a1c0000        0`00340000             MEM_FREE    PAGE_NOACCESS                      Free
+      7ffb`9a1c0000     7ffb`9a1c1000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [user32; "C:\Windows\System32\user32.dll"]
       7ffb`9a1c1000     7ffb`9a255000        0`00094000 MEM_IMAGE   MEM_COMMIT  PAGE_EXECUTE_READ                  Image      [user32; "C:\Windows\System32\user32.dll"]
       7ffb`9a255000     7ffb`9a277000        0`00022000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [user32; "C:\Windows\System32\user32.dll"]
       7ffb`9a277000     7ffb`9a279000        0`00002000 MEM_IMAGE   MEM_COMMIT  PAGE_READWRITE                     Image      [user32; "C:\Windows\System32\user32.dll"]
       7ffb`9a279000     7ffb`9a36c000        0`000f3000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [user32; "C:\Windows\System32\user32.dll"]
+      7ffb`9a36c000     7ffb`9ab30000        0`007c4000             MEM_FREE    PAGE_NOACCESS                      Free
+      7ffb`9ab30000     7ffb`9ab31000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [sechost; "C:\Windows\System32\sechost.dll"]
       7ffb`9ab31000     7ffb`9ab98000        0`00067000 MEM_IMAGE   MEM_COMMIT  PAGE_EXECUTE_READ                  Image      [sechost; "C:\Windows\System32\sechost.dll"]
       7ffb`9ab98000     7ffb`9abc0000        0`00028000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [sechost; "C:\Windows\System32\sechost.dll"]
       7ffb`9abc0000     7ffb`9abc4000        0`00004000 MEM_IMAGE   MEM_COMMIT  PAGE_READWRITE                     Image      [sechost; "C:\Windows\System32\sechost.dll"]
       7ffb`9abc4000     7ffb`9abce000        0`0000a000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [sechost; "C:\Windows\System32\sechost.dll"]
+      7ffb`9abce000     7ffb`9abd0000        0`00002000             MEM_FREE    PAGE_NOACCESS                      Free
+      7ffb`9abd0000     7ffb`9abd1000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [msvcrt; "C:\Windows\System32\msvcrt.dll"]
       7ffb`9abd1000     7ffb`9ac4a000        0`00079000 MEM_IMAGE   MEM_COMMIT  PAGE_EXECUTE_READ                  Image      [msvcrt; "C:\Windows\System32\msvcrt.dll"]
       7ffb`9ac4a000     7ffb`9ac64000        0`0001a000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [msvcrt; "C:\Windows\System32\msvcrt.dll"]
       7ffb`9ac64000     7ffb`9ac67000        0`00003000 MEM_IMAGE   MEM_COMMIT  PAGE_READWRITE                     Image      [msvcrt; "C:\Windows\System32\msvcrt.dll"]
       7ffb`9ac67000     7ffb`9ac69000        0`00002000 MEM_IMAGE   MEM_COMMIT  PAGE_WRITECOPY                     Image      [msvcrt; "C:\Windows\System32\msvcrt.dll"]
       7ffb`9ac69000     7ffb`9ac6c000        0`00003000 MEM_IMAGE   MEM_COMMIT  PAGE_READWRITE                     Image      [msvcrt; "C:\Windows\System32\msvcrt.dll"]
       7ffb`9ac6c000     7ffb`9ac73000        0`00007000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [msvcrt; "C:\Windows\System32\msvcrt.dll"]
+      7ffb`9ac73000     7ffb`9b290000        0`0061d000             MEM_FREE    PAGE_NOACCESS                      Free
+      7ffb`9b290000     7ffb`9b291000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [gdi32; "C:\Windows\System32\gdi32.dll"]
       7ffb`9b291000     7ffb`9b29f000        0`0000e000 MEM_IMAGE   MEM_COMMIT  PAGE_EXECUTE_READ                  Image      [gdi32; "C:\Windows\System32\gdi32.dll"]
       7ffb`9b29f000     7ffb`9b2b3000        0`00014000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [gdi32; "C:\Windows\System32\gdi32.dll"]
       7ffb`9b2b3000     7ffb`9b2b4000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READWRITE                     Image      [gdi32; "C:\Windows\System32\gdi32.dll"]
       7ffb`9b2b4000     7ffb`9b2b9000        0`00005000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [gdi32; "C:\Windows\System32\gdi32.dll"]
+      7ffb`9b2b9000     7ffb`9b2c0000        0`00007000             MEM_FREE    PAGE_NOACCESS                      Free
+      7ffb`9b2c0000     7ffb`9b2c1000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [msctf; "C:\Windows\System32\msctf.dll"]
       7ffb`9b2c1000     7ffb`9b3a1000        0`000e0000 MEM_IMAGE   MEM_COMMIT  PAGE_EXECUTE_READ                  Image      [msctf; "C:\Windows\System32\msctf.dll"]
       7ffb`9b3a1000     7ffb`9b3ca000        0`00029000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [msctf; "C:\Windows\System32\msctf.dll"]
       7ffb`9b3ca000     7ffb`9b3cd000        0`00003000 MEM_IMAGE   MEM_COMMIT  PAGE_READWRITE                     Image      [msctf; "C:\Windows\System32\msctf.dll"]
       7ffb`9b3cd000     7ffb`9b3de000        0`00011000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [msctf; "C:\Windows\System32\msctf.dll"]
+      7ffb`9b3de000     7ffb`9b420000        0`00042000             MEM_FREE    PAGE_NOACCESS                      Free
+      7ffb`9b420000     7ffb`9b421000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [imm32; "C:\Windows\System32\imm32.dll"]
       7ffb`9b421000     7ffb`9b440000        0`0001f000 MEM_IMAGE   MEM_COMMIT  PAGE_EXECUTE_READ                  Image      [imm32; "C:\Windows\System32\imm32.dll"]
       7ffb`9b440000     7ffb`9b447000        0`00007000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [imm32; "C:\Windows\System32\imm32.dll"]
       7ffb`9b447000     7ffb`9b448000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READWRITE                     Image      [imm32; "C:\Windows\System32\imm32.dll"]
       7ffb`9b448000     7ffb`9b451000        0`00009000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [imm32; "C:\Windows\System32\imm32.dll"]
+      7ffb`9b451000     7ffb`9b740000        0`002ef000             MEM_FREE    PAGE_NOACCESS                      Free
+      7ffb`9b740000     7ffb`9b741000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [ntdll; "C:\Windows\System32\ntdll.dll"]
       7ffb`9b741000     7ffb`9b86c000        0`0012b000 MEM_IMAGE   MEM_COMMIT  PAGE_EXECUTE_READ                  Image      [ntdll; "C:\Windows\System32\ntdll.dll"]
```

```
      7ffb`9b86c000     7ffb`9b8b4000        0`00048000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [ntdll; "C:\Windows\System32\ntdll.dll"]
      7ffb`9b8b4000     7ffb`9b8b5000        0`00001000 MEM_IMAGE   MEM_COMMIT  PAGE_READWRITE                     Image      [ntdll; "C:\Windows\System32\ntdll.dll"]
      7ffb`9b8b5000     7ffb`9b8b7000        0`00002000 MEM_IMAGE   MEM_COMMIT  PAGE_WRITECOPY                     Image      [ntdll; "C:\Windows\System32\ntdll.dll"]
      7ffb`9b8b7000     7ffb`9b8c0000        0`00009000 MEM_IMAGE   MEM_COMMIT  PAGE_READWRITE                     Image      [ntdll; "C:\Windows\System32\ntdll.dll"]
      7ffb`9b8c0000     7ffb`9b949000        0`00089000 MEM_IMAGE   MEM_COMMIT  PAGE_READONLY                      Image      [ntdll; "C:\Windows\System32\ntdll.dll"]
+     7ffb`9b949000     7fff`ffff0000        4`646a7000             MEM_FREE    PAGE_NOACCESS                      Free
```

Note the first no access region highlighted in red. It also includes a subregion to catch NULL pointer access. The regions highlighted in blue belong to the M1 module. The first read-only one belongs to MZ/PE header and the second one, execute-read, belongs to the code section. Another command variant shows a summary:

```
0:000> !address -summary

--- Usage Summary ---------------- RgnCount ----------- Total Size -------- %ofBusy %ofTotal
Free                                     67     7ffe`f8122000 ( 127.996 TB)           100.00%
<unknown>                                39        1`05947000 (   4.087 GB)  99.11%    0.00%
Image                                   162        0`01e0f000 (  30.059 MB)   0.71%    0.00%
Stack                                    12        0`00400000 (   4.000 MB)   0.09%    0.00%
Heap                                     12        0`001c3000 (   1.762 MB)   0.04%    0.00%
Other                                     7        0`001ac000 (   1.672 MB)   0.04%    0.00%
TEB                                       4        0`00008000 (  32.000 kB)   0.00%    0.00%
PEB                                       1        0`00001000 (   4.000 kB)   0.00%    0.00%

--- Type Summary (for busy) ------ RgnCount ----------- Total Size -------- %ofBusy %ofTotal
MEM_PRIVATE                              45        1`030dd000 (   4.048 GB)  98.15%    0.00%
MEM_MAPPED                               30        0`02fe2000 (  47.883 MB)   1.13%    0.00%
MEM_IMAGE                               162        0`01e0f000 (  30.059 MB)   0.71%    0.00%

--- State Summary ---------------- RgnCount ----------- Total Size -------- %ofBusy %ofTotal
MEM_FREE                                 67     7ffe`f8122000 ( 127.996 TB)           100.00%
MEM_RESERVE                              22        1`04488000 (   4.067 GB)  98.62%    0.00%
MEM_COMMIT                              215        0`03a46000 (  58.273 MB)   1.38%    0.00%

--- Protect Summary (for commit) - RgnCount ----------- Total Size -------- %ofBusy %ofTotal
PAGE_READONLY                           117        0`02789000 (  39.535 MB)   0.94%    0.00%
PAGE_EXECUTE_READ                        30        0`011a7000 (  17.652 MB)   0.42%    0.00%
PAGE_READWRITE                           56        0`00100000 (   1.000 MB)   0.02%    0.00%
PAGE_READWRITE | PAGE_GUARD               4        0`0000c000 (  48.000 kB)   0.00%    0.00%
PAGE_WRITECOPY                            8        0`0000a000 (  40.000 kB)   0.00%    0.00%

--- Largest Region by Usage ----------- Base Address -------- Region Size ----------
Free                                    180`32b00000     7e74`ca570000 ( 126.456 TB)
<unknown>                              7ff4`fd170000        1`00020000 (   4.000 GB)
Image                                  7ffb`99921000        0`00261000 (   2.379 MB)
Stack                                     7`ce800000        0`000fb000 (1004.000 kB)
Heap                                    180`2e726000        0`00099000 ( 612.000 kB)
Other                                   180`2e9c0000        0`00181000 (   1.504 MB)
TEB                                       7`ce6fb000        0`00002000 (   8.000 kB)
PEB                                       7`ce6fa000        0`00001000 (   4.000 kB)
```

7. Let’s dump M1 module header and see all these sections:

```
0:000> !dh 00007ff7`6af00000

File Type: EXECUTABLE IMAGE
FILE HEADER VALUES
    8664 machine (X64)
       7 number of sections
62C31CF5 time date stamp Mon Jul  4 18:01:41 2022

       0 file pointer to symbol table
       0 number of symbols
      F0 size of optional header
      22 characteristics
            Executable
            App can handle >2gb addresses

OPTIONAL HEADER VALUES
     20B magic #
   14.32 linker version
    D400 size of code
    F200 size of initialized data
       0 size of uninitialized data
    1748 address of entry point
    1000 base of code
         ----- new -----
00007ff76af00000 image base
    1000 section alignment
     200 file alignment
       2 subsystem (Windows GUI)
    6.00 operating system version
    0.00 image version
    6.00 subsystem version
   20000 size of image
     400 size of headers
       0 checksum
0000000000100000 size of stack reserve
0000000000001000 size of stack commit
0000000000100000 size of heap reserve
0000000000001000 size of heap commit
    8160  DLL characteristics
            High entropy VA supported
            Dynamic base
            NX compatible
            Terminal server aware
       0 [       0] address [size] of Export Directory
   17F0C [      3C] address [size] of Import Directory
   1D000 [    1D78] address [size] of Resource Directory
   1B000 [     F30] address [size] of Exception Directory
       0 [       0] address [size] of Security Directory
   1F000 [     660] address [size] of Base Relocation Directory
   169F0 [      70] address [size] of Debug Directory
       0 [       0] address [size] of Description Directory
       0 [       0] address [size] of Special Directory
       0 [       0] address [size] of Thread Storage Directory
   168B0 [     140] address [size] of Load Configuration Directory
       0 [       0] address [size] of Bound Import Directory
    F000 [     2E8] address [size] of Import Address Table Directory
       0 [       0] address [size] of Delay Import Directory
       0 [       0] address [size] of COR20 Header Directory
       0 [       0] address [size] of Reserved Directory


SECTION HEADER #1
   .text name
    D230 virtual size
    1000 virtual address
    D400 size of raw data
     400 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
60000020 flags
         Code
         (no align specified)
         Execute Read

SECTION HEADER #2
  .rdata name
    98AC virtual size
    F000 virtual address
    9A00 size of raw data
    D800 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
40000040 flags
         Initialized Data
         (no align specified)
         Read Only


Debug Directories(4)
       Type       Size     Address  Pointer
       cv           37       16de8    155e8   Format: RSDS, guid, 1, 
C:\AWMA3\M1\x64\Release\M1.pdb
       (   12)      14       16e20    15620
       (   13)     31c       16e34    15634
       (   14)       0           0        0

SECTION HEADER #3
   .data name
    1EC0 virtual size
   19000 virtual address
     C00 size of raw data
   17200 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
C0000040 flags
         Initialized Data
         (no align specified)
         Read Write

SECTION HEADER #4
  .pdata name
     F30 virtual size
   1B000 virtual address
    1000 size of raw data
   17E00 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
40000040 flags
         Initialized Data
         (no align specified)
         Read Only
```

```
SECTION HEADER #5
  _RDATA name
     15C virtual size
   1C000 virtual address
     200 size of raw data
   18E00 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
40000040 flags
         Initialized Data
         (no align specified)
         Read Only

SECTION HEADER #6
   .rsrc name
    1D78 virtual size
   1D000 virtual address
    1E00 size of raw data
   19000 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
40000040 flags
         Initialized Data
         (no align specified)
         Read Only

SECTION HEADER #7
  .reloc name
     660 virtual size
   1F000 virtual address
     800 size of raw data
   1AE00 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
42000040 flags
         Initialized Data
         Discardable
         (no align specified)
         Read Only
```

8. Now we look **Import Address Table** and compare with the previous exercise:

```
0:000> dps 00007ff7`6af00000+F000 L2E8/8
00007ff7`6af0f000  00007ffb`99cbe880 kernel32!LoadLibraryWStub
00007ff7`6af0f008  00007ffb`99cc3780 kernel32!WriteConsoleW
00007ff7`6af0f010  00007ffb`99cc2c50 kernel32!CloseHandle
00007ff7`6af0f018  00007ffb`99cc2ed0 kernel32!CreateFileW
00007ff7`6af0f020  00007ffb`99cc3310 kernel32!SetFilePointerEx
00007ff7`6af0f028  00007ffb`99cc36b0 kernel32!GetConsoleMode
00007ff7`6af0f030  00007ffb`99cc36c0 kernel32!GetConsoleOutputCP
00007ff7`6af0f038  00007ffb`99cc3030 kernel32!FlushFileBuffers
00007ff7`6af0f040  00007ffb`9b764830 ntdll!RtlReAllocateHeap
```

```
00007ff7`6af0f048  00007ffb`9b7673a0 ntdll!RtlSizeHeap
00007ff7`6af0f050  00007ffb`99cb6340 kernel32!GetProcessHeapStub
00007ff7`6af0f058  00007ffb`99cb9290 kernel32!LCMapStringWStub
00007ff7`6af0f060  00007ffb`99cbf4c0 kernel32!FlsFreeStub
00007ff7`6af0f068  00007ffb`99cba630 kernel32!FlsSetValueStub
00007ff7`6af0f070  00007ffb`99cb82b0 kernel32!FlsGetValueStub
00007ff7`6af0f078  00007ffb`99cc2a00 kernel32!RtlCaptureContext
00007ff7`6af0f080  00007ffb`99cc0b20 kernel32!RtlLookupFunctionEntryStub
00007ff7`6af0f088  00007ffb`99cc5ab0 kernel32!RtlVirtualUnwindStub
00007ff7`6af0f090  00007ffb`99cda370 kernel32!UnhandledExceptionFilterStub
00007ff7`6af0f098  00007ffb`99cbe6d0 kernel32!SetUnhandledExceptionFilterStub
00007ff7`6af0f0a0  00007ffb`99cc2bd0 kernel32!GetCurrentProcess
00007ff7`6af0f0a8  00007ffb`99cbf800 kernel32!TerminateProcessStub
00007ff7`6af0f0b0  00007ffb`99cbb7d0 kernel32!IsProcessorFeaturePresentStub
00007ff7`6af0f0b8  00007ffb`99cb6670 kernel32!QueryPerformanceCounterStub
00007ff7`6af0f0c0  00007ffb`99cc2be0 kernel32!GetCurrentProcessId
00007ff7`6af0f0c8  00007ffb`99ca6170 kernel32!GetCurrentThreadId
00007ff7`6af0f0d0  00007ffb`99cb7a90 kernel32!GetSystemTimeAsFileTimeStub
00007ff7`6af0f0d8  00007ffb`9b7b5c50 ntdll!RtlInitializeSListHead
00007ff7`6af0f0e0  00007ffb`99cbe730 kernel32!IsDebuggerPresentStub
00007ff7`6af0f0e8  00007ffb`99cbba00 kernel32!GetStartupInfoWStub
00007ff7`6af0f0f0  00007ffb`99cbb790 kernel32!GetModuleHandleWStub
00007ff7`6af0f0f8  00007ffb`99cbe2e0 kernel32!RtlUnwindExStub
00007ff7`6af0f100  00007ffb`99cb62e0 kernel32!GetLastErrorStub
00007ff7`6af0f108  00007ffb`99cb6360 kernel32!SetLastErrorStub
00007ff7`6af0f110  00007ffb`9b77a890 ntdll!RtlEnterCriticalSection
00007ff7`6af0f118  00007ffb`9b77b880 ntdll!RtlLeaveCriticalSection
00007ff7`6af0f120  00007ffb`9b75e430 ntdll!RtlDeleteCriticalSection
00007ff7`6af0f128  00007ffb`99cc2d50 kernel32!InitializeCriticalSectionAndSpinCount
00007ff7`6af0f130  00007ffb`99cbb880 kernel32!TlsAllocStub
00007ff7`6af0f138  00007ffb`99ca6160 kernel32!TlsGetValueStub
00007ff7`6af0f140  00007ffb`99cb6300 kernel32!TlsSetValueStub
00007ff7`6af0f148  00007ffb`99cbc0b0 kernel32!TlsFreeStub
00007ff7`6af0f150  00007ffb`99cba650 kernel32!FreeLibraryStub
00007ff7`6af0f158  00007ffb`99cb93b0 kernel32!GetProcAddressStub
00007ff7`6af0f160  00007ffb`99cb93f0 kernel32!LoadLibraryExWStub
00007ff7`6af0f168  00007ffb`9b7ba950 ntdll!RtlEncodePointer
00007ff7`6af0f170  00007ffb`99cbbe40 kernel32!RaiseExceptionStub
00007ff7`6af0f178  00007ffb`99cbb960 kernel32!RtlPcToFileHeaderStub
00007ff7`6af0f180  00007ffb`99cbb9c0 kernel32!GetStdHandleStub
00007ff7`6af0f188  00007ffb`99cc3360 kernel32!WriteFile
00007ff7`6af0f190  00007ffb`99cbbfa0 kernel32!GetModuleFileNameWStub
00007ff7`6af0f198  00007ffb`99cbc660 kernel32!ExitProcessImplementation
00007ff7`6af0f1a0  00007ffb`99cbe090 kernel32!GetModuleHandleExWStub
00007ff7`6af0f1a8  00007ffb`9b768e70 ntdll!RtlAllocateHeap
00007ff7`6af0f1b0  00007ffb`99cb5ef0 kernel32!HeapFreeStub
00007ff7`6af0f1b8  00007ffb`99cc2f30 kernel32!FindClose
00007ff7`6af0f1c0  00007ffb`99cc2f90 kernel32!FindFirstFileExW
00007ff7`6af0f1c8  00007ffb`99cc3000 kernel32!FindNextFileW
00007ff7`6af0f1d0  00007ffb`99cbe660 kernel32!IsValidCodePageStub
00007ff7`6af0f1d8  00007ffb`99cbe050 kernel32!GetACPStub
00007ff7`6af0f1e0  00007ffb`99cc0120 kernel32!GetOEMCPStub
00007ff7`6af0f1e8  00007ffb`99cbc960 kernel32!GetCPInfoStub
00007ff7`6af0f1f0  00007ffb`99cbe710 kernel32!GetCommandLineAStub
00007ff7`6af0f1f8  00007ffb`99cbd660 kernel32!GetCommandLineWStub
00007ff7`6af0f200  00007ffb`99cb5fc0 kernel32!MultiByteToWideCharStub
00007ff7`6af0f208  00007ffb`99cb6010 kernel32!WideCharToMultiByteStub
00007ff7`6af0f210  00007ffb`99cbe1f0 kernel32!GetEnvironmentStringsWStub
00007ff7`6af0f218  00007ffb`99cbe210 kernel32!FreeEnvironmentStringsWStub
00007ff7`6af0f220  00007ffb`99cbee30 kernel32!SetStdHandleStub
```

```
00007ff7`6af0f228  00007ffb`99cc3120 kernel32!GetFileType
00007ff7`6af0f230  00007ffb`99cbc9d0 kernel32!GetStringTypeWStub
00007ff7`6af0f238  00007ffb`99cbe7c0 kernel32!FlsAllocStub
00007ff7`6af0f240  00000000`00000000
00007ff7`6af0f248  00007ffb`9a1eb7b0 user32!PostQuitMessage
00007ff7`6af0f250  00007ffb`9a1f2180 user32!NtUserEndPaint
00007ff7`6af0f258  00007ffb`9a1f1dd0 user32!NtUserBeginPaint
00007ff7`6af0f260  00007ffb`9b7e3b80 ntdll!NtdllDefWindowProc_W
00007ff7`6af0f268  00007ffb`9a1f1fe0 user32!NtUserDestroyWindow
00007ff7`6af0f270  00007ffb`9a20fbd0 user32!DialogBoxParamW
00007ff7`6af0f278  00007ffb`9a1e9940 user32!UpdateWindow
00007ff7`6af0f280  00007ffb`91c85da0 apphelp!SrHook_ShowWindow
00007ff7`6af0f288  00007ffb`9a218bf0 user32!EndDialog
00007ff7`6af0f290  00007ffb`9a1c7bb0 user32!RegisterClassExW
00007ff7`6af0f298  00007ffb`9a1cb110 user32!LoadCursorW
00007ff7`6af0f2a0  00007ffb`9a1c9760 user32!LoadIconW
00007ff7`6af0f2a8  00007ffb`9a1d0bf0 user32!DispatchMessageW
00007ff7`6af0f2b0  00007ffb`9a1d63e0 user32!TranslateMessage
00007ff7`6af0f2b8  00007ffb`9a1e4ea0 user32!TranslateAcceleratorW
00007ff7`6af0f2c0  00007ffb`9a1e4620 user32!GetMessageW
00007ff7`6af0f2c8  00007ffb`9a1e7950 user32!LoadAcceleratorsW
00007ff7`6af0f2d0  00007ffb`9a1e9010 user32!LoadStringW
00007ff7`6af0f2d8  00007ffb`9a1c8030 user32!CreateWindowExW
00007ff7`6af0f2e0  00000000`00000000
```

Note that we have real addresses in the accessible memory.

9. Let’s now check how imported functions are called. We now get stack trace for the current thread:

```
0:000> k
# Child-SP          RetAddr               Call Site
00 00000007`ce55f8c8 00007ffb`9a1e464e     win32u!NtUserGetMessage+0x14
01 00000007`ce55f8d0 00007ff7`6af010ac     user32!GetMessageW+0x2e
02 00000007`ce55f930 00007ff7`6af016da     M1+0x10ac
03 00000007`ce55f9a0 00007ffb`99cb54e0     M1+0x16da
04 00000007`ce55f9e0 00007ffb`9b74485b     kernel32!BaseThreadInitThunk+0x10
05 00000007`ce55fa10 00000000`00000000     ntdll!RtlUserThreadStart+0x2b
```

Recall that a return address is a return address for the call site below, so its backward disassembly normally shows a call CPU instruction, this time we expect a call to *GetMessageW*:

```
0:000> ub 00007ff7`6af010ac
M1+0x1089:
00007ff7`6af01089 488b4c2470      mov     rcx,qword ptr [rsp+70h]
00007ff7`6af0108e ff1534e20000    call    qword ptr [M1+0xf2c8 (00007ff7`6af0f2c8)]
00007ff7`6af01094 4889442420      mov     qword ptr [rsp+20h],rax
00007ff7`6af01099 4533c9          xor     r9d,r9d
00007ff7`6af0109c 4533c0          xor     r8d,r8d
00007ff7`6af0109f 33d2            xor     edx,edx
00007ff7`6af010a1 488d4c2428      lea     rcx,[rsp+28h]
00007ff7`6af010a6 ff1514e20000    call    qword ptr [M1+0xf2c0 (00007ff7`6af0f2c0)]
```

Square brackets mean an indirect address. The value at memory address **00007ff7`6af0f2c0** should contain an address to transfer execution:

```
0:000> dps 00007ff7`6af0f2c0 L1
00007ff7`6af0f2c0  00007ffb`9a1e4620 user32!GetMessageW
```

Note that the address **00007ff7`6af0f2c0** is inside **Import Address Table** above.

10. Finally, we check the integrity of our M1 module:

```
0:000> !chkimg -v -d M1
Searching for module with expression: M1
Error for M1: Could not find image file for the module. Make sure binaries are included in the
symbol path.
```

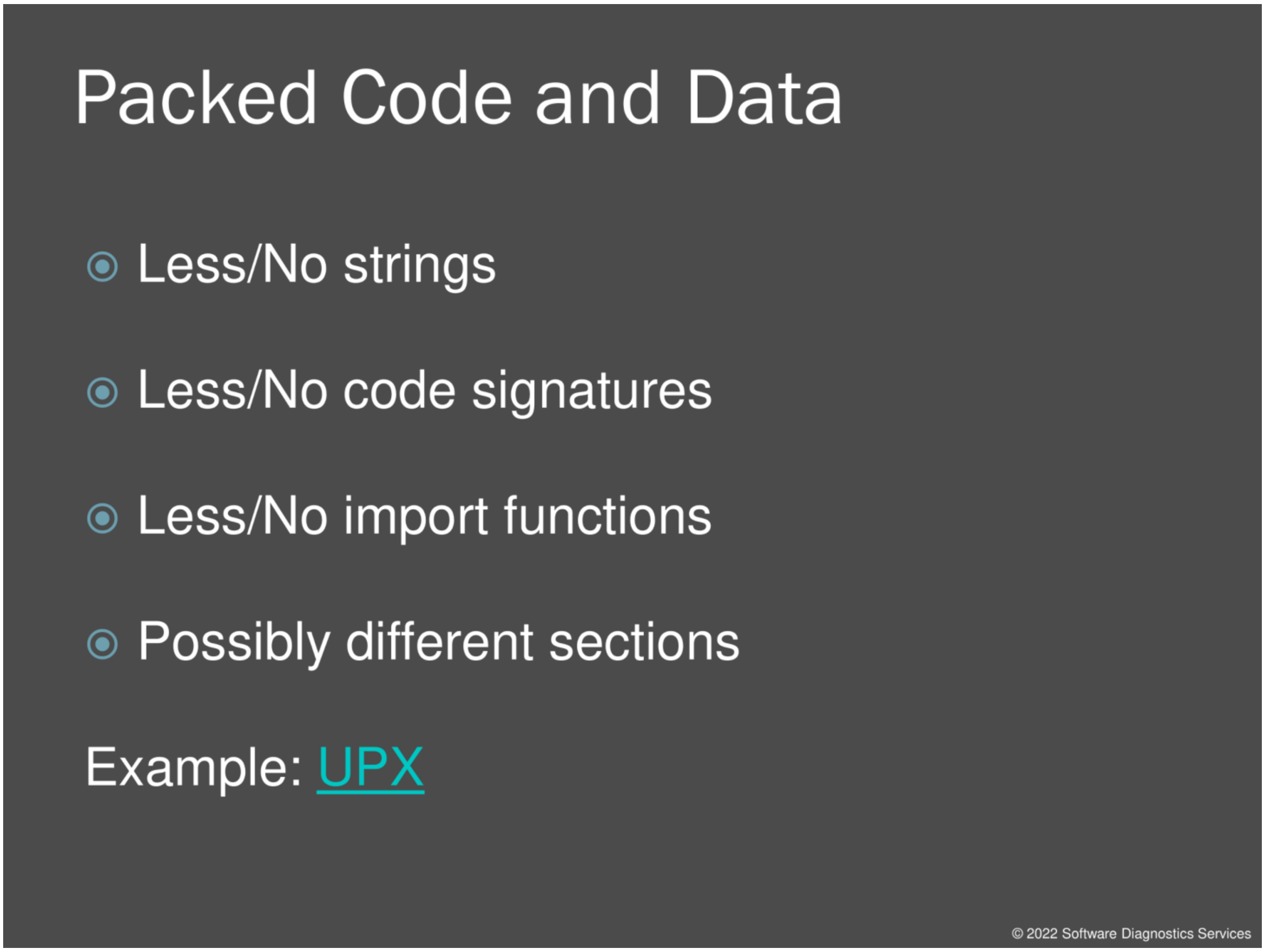
WinDbg Preview cannot find a module to compare what's inside a dump file. So we specify an executable search path:

```
0:000> .exepath+ C:\AWMA-Dumps\Executables\
Executable image search path is: srv*;C:\AWMA-Dumps\Executables\
Expanded Executable image search path is:
SRV*c:\mss*https://msdl.microsoft.com/download/symbols;c:\awma-dumps\executables\

************* Symbol Path validation summary **************
Response                         Time (ms)     Location
Deferred                                       srv*
OK                                             C:\AWMA-Dumps\Executables\

0:000> !chkimg -v -d M1
Searching for module with expression: M1
Will apply relocation fixups to file used for comparison
Will ignore NOP/LOCK errors
Will ignore patched instructions
Image specific ignores will be applied
Comparison image path: C:\AWMA-Dumps\Executables\M1.exe
No range specified

Scanning section:    .text
Size: 53808
Range to scan: 7ff76af01000-7ff76af0e230
Total bytes compared: 53808(100%)
Number of errors: 0

Scanning section:   .rdata
Size: 39084
Range to scan: 7ff76af0f000-7ff76af188ac
Total bytes compared: 39084(100%)
Number of errors: 0

Scanning section:   .pdata
Size: 3888
Range to scan: 7ff76af1b000-7ff76af1bf30
Total bytes compared: 3888(100%)
Number of errors: 0

Scanning section:   _RDATA
Size: 348
Range to scan: 7ff76af1c000-7ff76af1c15c
Total bytes compared: 348(100%)
Number of errors: 0

Scanning section:    .rsrc
Size: 7544
Range to scan: 7ff76af1d000-7ff76af1ed78
Total bytes compared: 7544(100%)
```

```
Number of errors: 0
0 errors : M1
```

11. Close the log file:

```
0:000> .logclose
Closing open log file C:\AWMA-Dumps\M1B.log
```

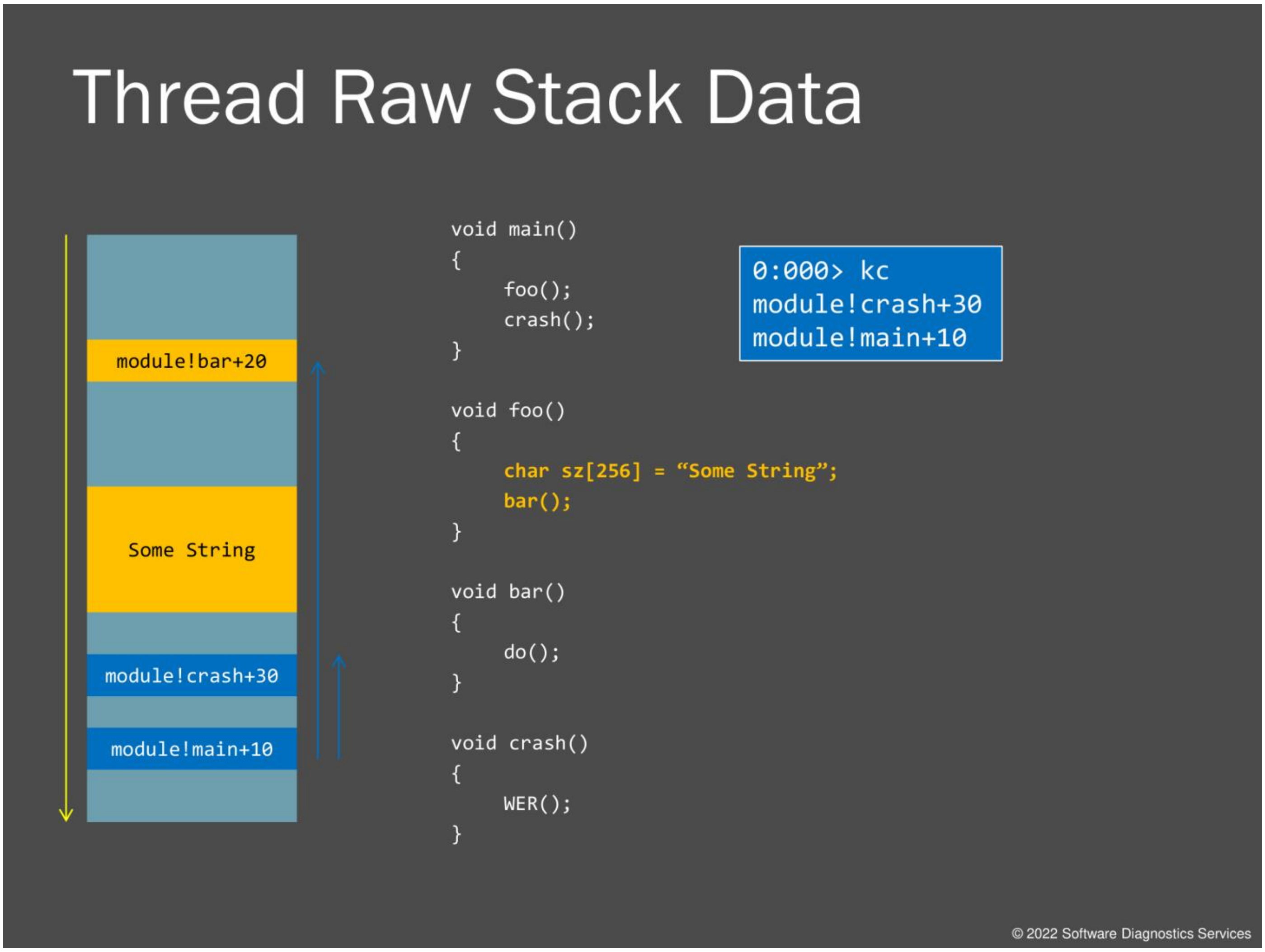


Packed Code and Data

- Less/No strings
- Less/No code signatures
- Less/No import functions
- Possibly different sections

Example: UPX

The sections and their names can be arbitrary. It is possible to have a different name and even one or two sections only. In the end, a module is just a binary that can be loaded at some memory address. It is even possible to write your own loader and linker. Code and data may also be packed. Here we look at a process dump file that contains packed modules. One module after compilation was packed by UPX packer, and upon start, a program loads it and also loads the same module but unpacked for comparison. Usually, if you search for strings in any normal module, you find plenty of them. Obviously, you find fewer of them in a packed module, although some fragments may survive (the so-called **Pre-Obfuscation Residue** pattern). Every function usually has some standard signatures, such as the so-called function prolog and epilog that have the same binary values. Also, Import Address Table might be empty or contain a few specific functions, and section names and attributes may be completely different, as in the case of UPX (https://upx.github.io/).



Thread Raw Stack Data

module!bar+20

Some String

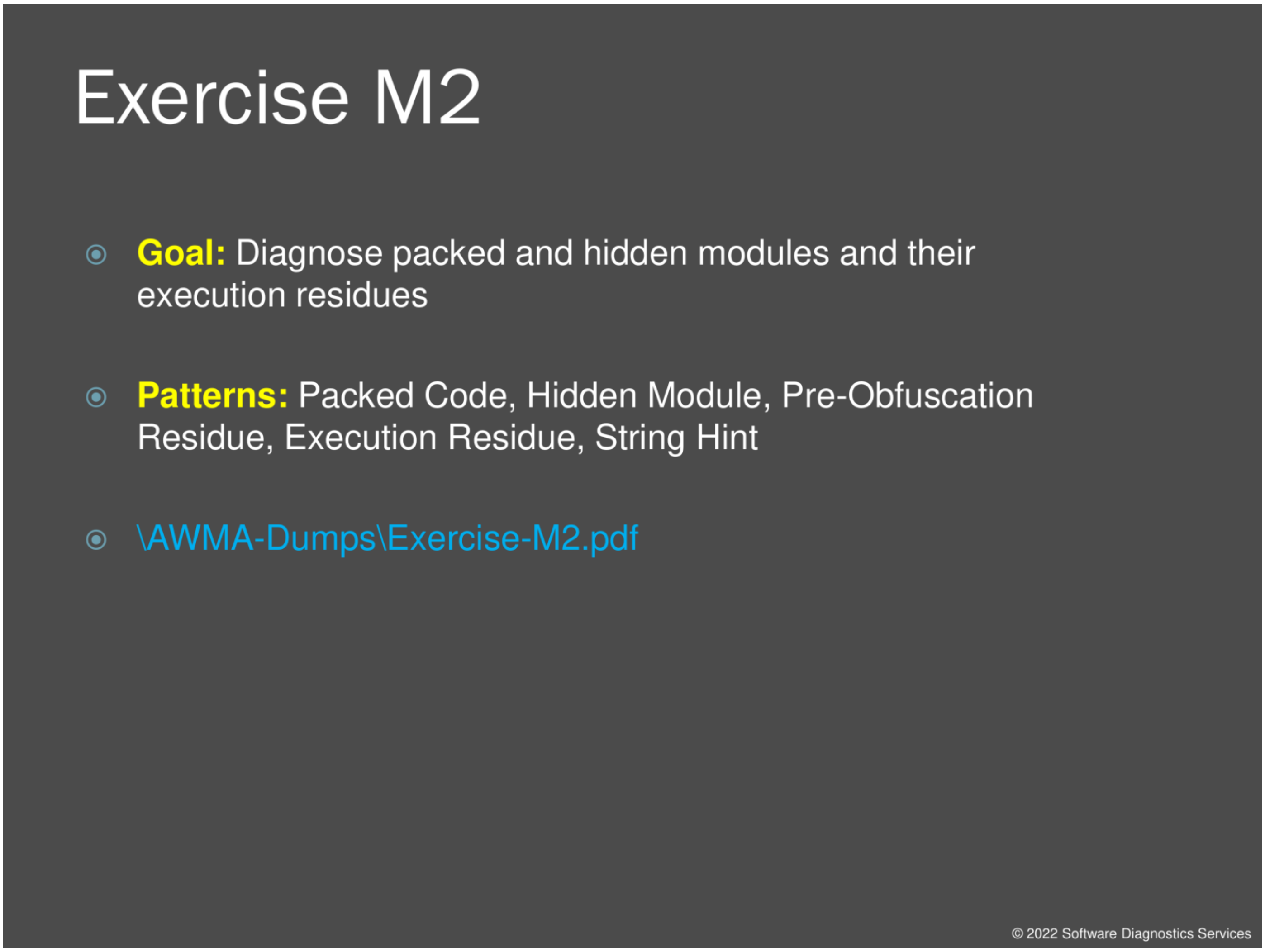
module!crash+30

module!main+10

```
void main()
{
     foo();
     crash();
}

void foo()
{
     char sz[256] = “Some String”;
     bar();
}

void bar()
{
     do();
}

void crash()
{
     WER();
}
```

```
0:000> kc
module!crash+30
module!main+10
```

Please recall that each thread of execution has its own region in user space called a stack. We also call it a raw stack to differentiate it from a stack trace. Every function call results in a return address stored there. Sometimes such return addresses are overwritten by subsequent execution, and sometimes they survive. We call this **Execution Residue** pattern. We can also see ASCII and UNICODE strings if they survive there. For example, after the *crash()* function execution that calls exception processing code, we see a stack trace, but there is also surviving execution residue of the *bar()* function because of a pre-allocated buffer. Please also note that a stack grows towards lower addresses during function calls, as shown by blue arrows on the right of the raw stack box.



Exercise M2

- **Goal:** Diagnose packed and hidden modules and their execution residues
- **Patterns:** Packed Code, Hidden Module, Pre-Obfuscation Residue, Execution Residue, String Hint
- \AWMA-Dumps\Exercise-M2.pdf

Exercise M2

Goal: Diagnose packed and hidden modules and their execution residues.

Patterns: Packed Code, Hidden Module, Pre-Obfuscation Residue, Execution Residue.

1. Launch WinDbg Preview.

2. Open \AWMA-Dumps\Processes\M2.dmp.

3. We get the dump file loaded:

```
Microsoft (R) Windows Debugger Version 10.0.25136.1001 X86
Copyright (c) Microsoft Corporation. All rights reserved.


Loading Dump File [C:\AWMA-Dumps\Processes\M2.DMP]
User Mini Dump File with Full Memory: Only application data is available


************* Path validation summary **************
Response                         Time (ms)     Location
Deferred                                       srv*
Symbol search path is: srv*
Executable search path is:
Windows 7 Version 7601 (Service Pack 1) MP (4 procs) Free x86 compatible
Product: WinNt, suite: SingleUserTS Personal
Machine Name:
Debug session time: Wed Jan 30 19:24:22.000 2013 (UTC + 1:00)
System Uptime: 21 days 7:17:59.279
Process Uptime: 0 days 0:00:28.000
.......
For analysis of this file, run !analyze -v
eax=00000000 ebx=00000000 ecx=00000000 edx=00000000 esi=0045f9bc edi=00000000
eip=76fffd71 esp=0045f978 ebp=0045f9e0 iopl=0         nv up ei pl zr na pe nc
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00000246
ntdll!NtDelayExecution+0x15:
76fffd71 83c404          add     esp,4
```

4. Open a log file:

```
0:000> .logopen C:\AWMA-Dumps\M2.log
Opened log file 'C:\AWMA-Dumps\M2.log'
```

5. List modules and their timestamps:

```
0:000> lmt
start    end        module name
012e0000 012ec000   M2         Wed Jan 30 18:23:18 2013 (51096516)
5ea70000 5eb46000   msvcr110   Tue Nov 06 03:35:42 2012 (5098858E)
60640000 60654000   calc3du    Wed Jan 30 16:21:24 2013 (51094884)
71610000 71627000   calc3d     Wed Jan 30 16:21:24 2013 (51094884)
751e0000 752f0000   kernel32   Mon Aug 20 18:40:01 2012 (50327671)
75390000 753d7000   KERNELBASE  Mon Aug 20 18:40:02 2012 (50327672)
76fe0000 77160000   ntdll      Thu Nov 17 05:28:47 2011 (4EC49B8F)
```

Note that some modules have approximately the same build timestamp and, therefore, can be related.

6. Let’s check headers for each module. We can use **!for_each_module** command to automate this task (here logs are useful):

```
0:000> !for_each_module ".echo Module name: @#ModuleName; !dh @#ModuleName"
[...]

Module name: calc3du

File Type: DLL
FILE HEADER VALUES
     14C machine (i386)
       5 number of sections
51094884 time date stamp Wed Jan 30 16:21:24 2013

       0 file pointer to symbol table
       0 number of symbols
      E0 size of optional header
    2102 characteristics
            Executable
            32 bit word machine
            DLL

OPTIONAL HEADER VALUES
     10B magic #
   11.00 linker version
    6400 size of code
    9800 size of initialized data
       0 size of uninitialized data
    1262 address of entry point
    1000 base of code
         ----- new -----
60640000 image base
    1000 section alignment
     200 file alignment
       2 subsystem (Windows GUI)
    6.00 operating system version
    0.00 image version
    6.00 subsystem version
   14000 size of image
     400 size of headers
       0 checksum
00100000 size of stack reserve
00001000 size of stack commit
00100000 size of heap reserve
00001000 size of heap commit
     140  DLL characteristics
            Dynamic base
            NX compatible
    C600 [      A9] address [size] of Export Directory
    C034 [      28] address [size] of Import Directory
   10000 [     1E0] address [size] of Resource Directory
       0 [       0] address [size] of Exception Directory
       0 [       0] address [size] of Security Directory
   11000 [     B80] address [size] of Base Relocation Directory
    8140 [      38] address [size] of Debug Directory
       0 [       0] address [size] of Description Directory
       0 [       0] address [size] of Special Directory
       0 [       0] address [size] of Thread Storage Directory
    BCA0 [      40] address [size] of Load Configuration Directory
       0 [       0] address [size] of Bound Import Directory
    8000 [     100] address [size] of Import Address Table Directory
       0 [       0] address [size] of Delay Import Directory
       0 [       0] address [size] of COR20 Header Directory
       0 [       0] address [size] of Reserved Directory


SECTION HEADER #1
   .text name
    6320 virtual size
    1000 virtual address
    6400 size of raw data
     400 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
60000020 flags
         Code
         (no align specified)
         Execute Read

SECTION HEADER #2
  .rdata name
    46A9 virtual size
    8000 virtual address
    4800 size of raw data
    6800 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
40000040 flags
         Initialized Data
         (no align specified)
         Read Only


Debug Directories(2)
       Type        Size      Address  Pointer
       cv            3b         bce8     a4e8    Format: RSDS, guid, 1,
C:\Work\AWMA\M2\Release\calc3d.pdb
       (    12)       10         bd24     a524

SECTION HEADER #3
   .data name
    2BF4 virtual size
    D000 virtual address
     E00 size of raw data
    B000 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
C0000040 flags
         Initialized Data
         (no align specified)
         Read Write

SECTION HEADER #4
   .rsrc name
     1E0 virtual size
   10000 virtual address
     200 size of raw data
    BE00 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
40000040 flags
         Initialized Data
         (no align specified)
         Read Only

SECTION HEADER #5
  .reloc name
    2106 virtual size
   11000 virtual address
    2200 size of raw data
    C000 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
42000040 flags
         Initialized Data
         Discardable
         (no align specified)
         Read Only
Module name: calc3d

File Type: DLL
FILE HEADER VALUES
     14C machine (i386)
       3 number of sections
51094884 time date stamp Wed Jan 30 16:21:24 2013

       0 file pointer to symbol table
       0 number of symbols
      E0 size of optional header
    2102 characteristics
            Executable
            32 bit word machine
            DLL

OPTIONAL HEADER VALUES
     10B magic #
   11.00 linker version
    6000 size of code
    1000 size of initialized data
    F000 size of uninitialized data
   15600 address of entry point
   10000 base of code
         ----- new -----
71610000 image base
    1000 section alignment
     200 file alignment
```

```
       2 subsystem (Windows GUI)
    6.00 operating system version
    0.00 image version
    6.00 subsystem version
   17000 size of image
    1000 size of headers
       0 checksum
00100000 size of stack reserve
00001000 size of stack commit
00100000 size of heap reserve
00001000 size of heap commit
     140  DLL characteristics
            Dynamic base
            NX compatible
   16274 [      AC] address [size] of Export Directory
   161DC [      98] address [size] of Import Directory
   16000 [     1DC] address [size] of Resource Directory
       0 [       0] address [size] of Exception Directory
       0 [       0] address [size] of Security Directory
   16320 [      10] address [size] of Base Relocation Directory
       0 [       0] address [size] of Debug Directory
       0 [       0] address [size] of Description Directory
       0 [       0] address [size] of Special Directory
       0 [       0] address [size] of Thread Storage Directory
   157CC [      48] address [size] of Load Configuration Directory
       0 [       0] address [size] of Bound Import Directory
       0 [       0] address [size] of Import Address Table Directory
       0 [       0] address [size] of Delay Import Directory
       0 [       0] address [size] of COR20 Header Directory
       0 [       0] address [size] of Reserved Directory

SECTION HEADER #1
    UPX0 name
    F000 virtual size
    1000 virtual address
       0 size of raw data
       0 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
60000080 flags
         Uninitialized Data
         (no align specified)
         Execute Read

SECTION HEADER #2
    UPX1 name
    6000 virtual size
   10000 virtual address
    5A00 size of raw data
     400 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
60000040 flags
         Initialized Data
         (no align specified)
         Execute Read
```

```
SECTION HEADER #3
   .rsrc name
    1000 virtual size
   16000 virtual address
     400 size of raw data
    5E00 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
C0000040 flags
         Initialized Data
         (no align specified)
         Read Write

[...]
```

Note that we see *calc3d.dll* loaded at **71610000** and having an empty **Import Address Table** and different section names **UPX0** and **UPX1**.

7. We now search UPX1 address range for ASCII strings (we can use **s-su** to search for UNICODE strings):

```
0:000> s-sa 71610000+10000 L6000
71624009  "GetCommandLineA"
7162401a  "GetCurrentThreadId"
7162402e  "IsDebuggerPresent"
71624041  "EncodePointer"
71624050  "DecodePointer"
7162405f  "IsProcessorFeaturePresent"
7162407a  "GetLastError"
71624088  "SetLastError"
71624096  "InterlockedIncrement"
716240ac  "InterlockedDecrement"
716240c2  "ExitProcess"
716240cf  "GetModuleHandleExW"
716240e3  "GetProcAddress"
716240f3  "MultiByteToWideChar"
71624108  "GetProcessHeap"
71624118  "GetStdHandle"
71624126  "GetFileType"
71624133  "InitializeCriticalSectionAndSpin"
71624153  "Count"
7162415a  "DeleteCriticalSection"
71624171  "GetStartupInfoW"
71624182  "GetModuleFileNameA"
71624196  "HeapFree"
716241a0  "QueryPerformanceCounter"
716241b9  "GetCurrentProcessId"
716241ce  "GetSystemTimeAsFileTime"
716241e7  "GetEnvironmentStringsW"
716241ff  "FreeEnvironmentStringsW"
71624218  "WideCharToMultiByte"
7162422d  "UnhandledExceptionFilter"
71624247  "SetUnhandledExceptionFilter"
71624264  "GetCurrentProcess"
71624277  "TerminateProcess"
71624289  "TlsAlloc"
71624293  "TlsGetValue"
```

```
716242a0  "TlsSetValue"
716242ad  "TlsFree"
716242b6  "GetModuleHandleW"
716242c8  "Sleep"
716242cf  "EnterCriticalSection"
716242e5  "LeaveCriticalSection"
716242fb  "IsValidCodePage"
7162430c  "GetACP"
71624314  "GetOEMCP"
7162431e  "GetCPInfo"
71624329  "WriteFile"
71624334  "GetModuleFileNameW"
71624348  "LoadLibraryExW"
71624358  "RtlUnwind"
71624363  "HeapAlloc"
7162436e  "HeapReAlloc"
7162437b  "GetStringTypeW"
7162438b  "OutputDebugStringW"
7162439f  "LoadLibraryW"
716243ad  "HeapSize"
716243b7  "LCMapStringW"
716243c5  "FlushFileBuffers"
716243d7  "GetConsoleCP"
716243e5  "GetConsoleMode"
716243f5  "SetStdHandle"
71624403  "SetFilePointerEx"
71624415  "WriteConsoleW"
71624424  "CloseHandle"
71624431  "CreateFileW"
[...]
71624ae4  ".text"
71624b0b  "`.rdata"
71624b33  "@.data"
71624b5c  ".rsrc"
71624b83  "@.reloc"
[...]
71624ce7  "o:\Work\AWMA\M2\ReleaseN"
71624d02  "\:c3d.pd"
[...]
7162502a  "ommand"
71625041  "IsDe"
71625049  "buggerP"
71625054  "Encodnmk"
[...]
```

8. Now we check the number of threads and look at the current thread raw stack:

```
0:000> ~
.  0  Id: 233c.1254 Suspend: 0 Teb: 7efdd000 Unfrozen

0:000> k
# ChildEBP RetAddr
00 0045f978 753a3bc8     ntdll!NtDelayExecution+0x15
01 0045f9e0 753a4498     KERNELBASE!SleepEx+0x65
*** WARNING: Unable to verify checksum for M2.exe
02 0045f9f0 012e101e     KERNELBASE!Sleep+0xf
WARNING: Stack unwind information not available. Following frames may be wrong.
03 0045fa38 751f33aa     M2+0x101e
04 0045fa44 77019ef2     kernel32!BaseThreadInitThunk+0xe
```

```
05 0045fa84 77019ec5     ntdll!__RtlUserThreadStart+0x70
06 0045fa9c 00000000     ntdll!_RtlUserThreadStart+0x1b
```

To get raw stack region boundaries we use **!teb** command:

```
0:000> !teb
TEB at 7efdd000
    ExceptionList:        0045f9d0
    StackBase:            00460000
    StackLimit:           0045e000
    SubSystemTib:         00000000
    FiberData:            00001e00
    ArbitraryUserPointer: 00000000
    Self:                 7efdd000
    EnvironmentPointer:   00000000
    ClientId:             0000233c . 00001254
    RpcHandle:            00000000
    Tls Storage:          7efdd02c
    PEB Address:          7efde000
    LastErrorValue:       0
    LastStatusValue:      c0000139
    Count Owned Locks:    0
    HardErrorMode:        0
```

Now can dumps memory values with corresponding symbols using **dps** command:

```
0:000> dps 0045e000 00460000
0045e000  00000000
0045e004  00000000
0045e008  00000000
0045e00c  00000000
0045e010  00000000
[...]
0045eb80  00000000
0045eb84  00000000
0045eb88  0045ec18
0045eb8c  0045ebc4
0045eb90  753bea9e KERNELBASE!LCMapStringEx+0x130
0045eb94  00000000
0045eb98  00000200
0045eb9c  0045ee28
0045eba0  00000100
0045eba4  0045ec18
0045eba8  008a4498
0045ebac  7efb0222
0045ebb0  00000100
0045ebb4  0045ebe8
0045ebb8  753c0c6e KERNELBASE!WideCharToMultiByte+0x19f
0045ebbc  008a4498
0045ebc0  0045ec18
0045ebc4  0045ee18
0045ebc8  0045f2d0
0045ebcc  0045f3d0
0045ebd0  00000000
0045ebd4  00000100
0045ebd8  0045ec18
0045ebdc  0045ee28
0045ebe0  008a4498
0045ebe4  00000000
```

```
0045ebe8  0045f040
0045ebec  6064503b calc3du!fncalc3d+0x3fdb
0045ebf0  00000000
0045ebf4  00000000
0045ebf8  0045ec18
0045ebfc  00000001
0045ec00  0045f2d0
0045ec04  0045f040
0045ec08  6064504a calc3du!fncalc3d+0x3fea
0045ec0c  0045ee28
0045ec10  0000cccc
[...]
0045f964  0045f950
0045f968  7701c439 ntdll!LdrpLoadDll+0x635
0045f96c  0045fa28
0045f970  770571d5 ntdll!_except_handler4
0045f974  6db8e6f2
0045f978  76fffd71 ntdll!NtDelayExecution+0x15
0045f97c  753a3bc8 KERNELBASE!SleepEx+0x65
0045f980  00000000
0045f984  0045f9bc
0045f988  4f2ad6dc
0045f98c  00000000
0045f990  00000001
0045f994  00000000
0045f998  00000024
0045f99c  00000001
0045f9a0  00000000
0045f9a4  00000000
0045f9a8  00000000
0045f9ac  00000000
0045f9b0  00000000
0045f9b4  00000000
0045f9b8  00000000
0045f9bc  00000000
0045f9c0  80000000
0045f9c4  00000000
0045f9c8  0045f988
0045f9cc  001c001a
0045f9d0  0045fa28
0045f9d4  753c6fa0 KERNELBASE!_except_handler4
0045f9d8  3a53a6c4
0045f9dc  00000000
0045f9e0  0045f9f0
0045f9e4  753a4498 KERNELBASE!Sleep+0xf
0045f9e8  ffffffff
0045f9ec  00000000
0045f9f0  0045fa38
0045f9f4  012e101e M2+0x101e
0045f9f8  ffffffff
0045f9fc  012e1231 M2+0x1231
0045fa00  00000001
0045fa04  008c9660
0045fa08  008cbb78
0045fa0c  22fb0166
0045fa10  00000000
0045fa14  00000000
0045fa18  7efde000
0045fa1c  00000000
0045fa20  0045fa0c
```

```
0045fa24  000002c5
0045fa28  0045fa74
0045fa2c  012e17e9 M2+0x17e9
0045fa30  2390dab6
0045fa34  00000000
0045fa38  0045fa44
0045fa3c  751f33aa kernel32!BaseThreadInitThunk+0xe
0045fa40  7efde000
0045fa44  0045fa84
0045fa48  77019ef2 ntdll!__RtlUserThreadStart+0x70
0045fa4c  7efde000
0045fa50  1afddd0e
0045fa54  00000000
0045fa58  00000000
0045fa5c  7efde000
0045fa60  00000000
0045fa64  00000000
0045fa68  00000000
0045fa6c  0045fa50
0045fa70  00000000
0045fa74  ffffffff
0045fa78  770571d5 ntdll!_except_handler4
0045fa7c  6db8e2ba
0045fa80  00000000
0045fa84  0045fa9c
0045fa88  77019ec5 ntdll!_RtlUserThreadStart+0x1b
0045fa8c  012e1299 M2+0x1299
0045fa90  7efde000
0045fa94  00000000
0045fa98  00000000
0045fa9c  00000000
[...]
```

We see calc3du module residue and check if it is not coincidental such as a constant that falls into some module address range:

```
0:000> ub 6064504a
calc3du!fncalc3d+0x3fd2:
60645032 ff7524          push    dword ptr [ebp+24h]
60645035 ff156c806460    call    dword ptr [calc3du!fncalc3d+0x700c (6064806c)]
6064503b 8bf8            mov     edi,eax
6064503d 56              push    esi
6064503e e860000000      call    calc3du!fncalc3d+0x4043 (606450a3)
60645043 59              pop     ecx
60645044 53              push    ebx
60645045 e859000000      call    calc3du!fncalc3d+0x4043 (606450a3)

0:000> ub 6064503b
calc3du!fncalc3d+0x3fc7:
60645027 eb06            jmp     calc3du!fncalc3d+0x3fcf (6064502f)
60645029 ff7520          push    dword ptr [ebp+20h]
6064502c ff751c          push    dword ptr [ebp+1Ch]
6064502f 57              push    edi
60645030 56              push    esi
60645031 50              push    eax
60645032 ff7524          push    dword ptr [ebp+24h]
60645035 ff156c806460    call    dword ptr [calc3du!fncalc3d+0x700c (6064806c)]
```

Because the preceding instruction is a *call,* there is a much higher probability that this return address was saved during past execution. We can also check for strings in that region **s-sa** and **s-su** commands or interpret every value as a pointer to a string by using **dpa** and **dpu** commands. **dpp** command would treat every value as a memory address and show a value it points to together with possible symbols (double redirection).

9. We now check the whole modules *calc3d* and *calc3du* address ranges for any malicious **String Hints** such as website, password and HTTP forms:

```
0:000> lm
start    end        module name
012e0000 012ec000   M2       C (no symbols)
5ea70000 5eb46000   msvcr110   (deferred)
60640000 60654000   calc3du  C (export symbols)       calc3du.dll
71610000 71627000   calc3d     (deferred)
751e0000 752f0000   kernel32   (pdb symbols)
C:\WinDbg.Docker.AWMA\mss\wkernel32.pdb\E1C01974DA974A699700CC37CD94A9202\wkernel32.pdb
75390000 753d7000   KERNELBASE   (pdb symbols)
C:\WinDbg.Docker.AWMA\mss\wkernelbase.pdb\615FE84E96114FE8B63193C923E026F51\wkernelbase.pdb
76fe0000 77160000   ntdll      (pdb symbols)
C:\WinDbg.Docker.AWMA\mss\wntdll.pdb\D74F79EB1F8D4A45ABCD2F476CCABACC2\wntdll.pdb
```

Note: We see *C:\WinDbg.Docker.AWMA\mss* paths because when preparing these exercises we ran **.sympath+** *C:\WinDbg.Docker.AWMA\mss* after loading the dump to save downloaded symbol files to a docker image build folder. On your system, you may have *C:\ProgramData\Dbg\sym* as your downloaded symbol files folder.

```
0:000> s-su 60640000 60654000
[...]
60648178  https://www.dumpanalysis.com
[...]

0:000> s-su 71610000 71627000
[...]
71618178  https://www.dumpanalysis.com
[...]
```

10. Let's now check if there are any **Hidden Modules** not shown in the loaded module list by using the **.imgscan** command that searches for MZ/PE signatures:

```
0:000> .imgscan
MZ at 012e0000, prot 00000002, type 01000000 - size c000
  Name: M2.exe
MZ at 5ea70000, prot 00000002, type 01000000 - size d6000
  Name: MSVCR110.dll
MZ at 60640000, prot 00000002, type 01000000 - size 14000
  Name: calc3d.dll
MZ at 71610000, prot 00000002, type 01000000 - size 17000
  Name: calc3d.dll
MZ at 72e00000, prot 00000002, type 01000000 - size 8000
  Name: wow64cpu.dll
MZ at 72e10000, prot 00000002, type 01000000 - size 5c000
  Name: wow64win.dll
MZ at 72e70000, prot 00000002, type 01000000 - size 3f000
  Name: wow64.dll
MZ at 751e0000, prot 00000002, type 01000000 - size 110000
  Name: KERNEL32.dll
MZ at 75390000, prot 00000002, type 01000000 - size 47000
  Name: KERNELBASE.dll
```

```
MZ at 76e00000, prot 00000002, type 01000000 - size 1a9000
  Name: ntdll.dll
MZ at 76fe0000, prot 00000002, type 01000000 - size 180000
  Name: ntdll.dll
```

Note: *wow64* modules and two *ndll* modules can be explained by the fact that this 32-bit dump came from x64 Windows.

Let's double check these findings by searching for MZ strings. By default **s-sa** command ignores 2 byte ASCII sequences so we need to specify *l2* parameter. For example, seaching in M2 module address range reveals a second MZ/PE header and closest strings point to it being packed by UPX packer:

```
0:000> s -[l2]sa 012e0000 012ec000
012e0000  "MZ"
012e004d  "!This program cannot be run in D"
012e006d  "OS mode."
012e00c0  "S;"
012e00c8  "S;"
012e00d8  "S;"
012e00e0  "Rich"
012e00f0  "PE"
012e0170  "D""
012e017c  "0d"
012e01b8  "8!"
012e01e8  ".text"
012e020f  "`.rdata"
012e0237  "@.data"
012e0260  ".rsrc"
012e0268  "0d"
012e0287  "@.reloc"
012e1002  "!."
012e1008  " ."
012e100d  "!."
012e1013  " ."
[...]
012e40b0  "MZ"
012e40fd  "!This program cannot be run in D"
012e411d  "OS mode."
012e4188  "Rich"
012e4198  "PE"
012e4210  "tb"
012e4238  " c"
012e4290  "UPX0"
012e42b8  "UPX1"
012e42e0  ".rsrc"
012e448b  "3.08"
012e4490  "UPX!"
[...]
```

Dumping M2 module header shows this hidden module is located inside a resource section:

```
0:000> !dh 012e0000

[...]

SECTION HEADER #4
   .rsrc name
```

```
    6430 virtual size
    4000 virtual address
    6600 size of raw data
    1600 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
40000040 flags
         Initialized Data
         (no align specified)
         Read Only

[...]
```

If we dump ASCII strings we don't find many because the module was packed and not yet loaded for execution. However, we see some **Pre-Obfuscation Residue**, fragments of strings:

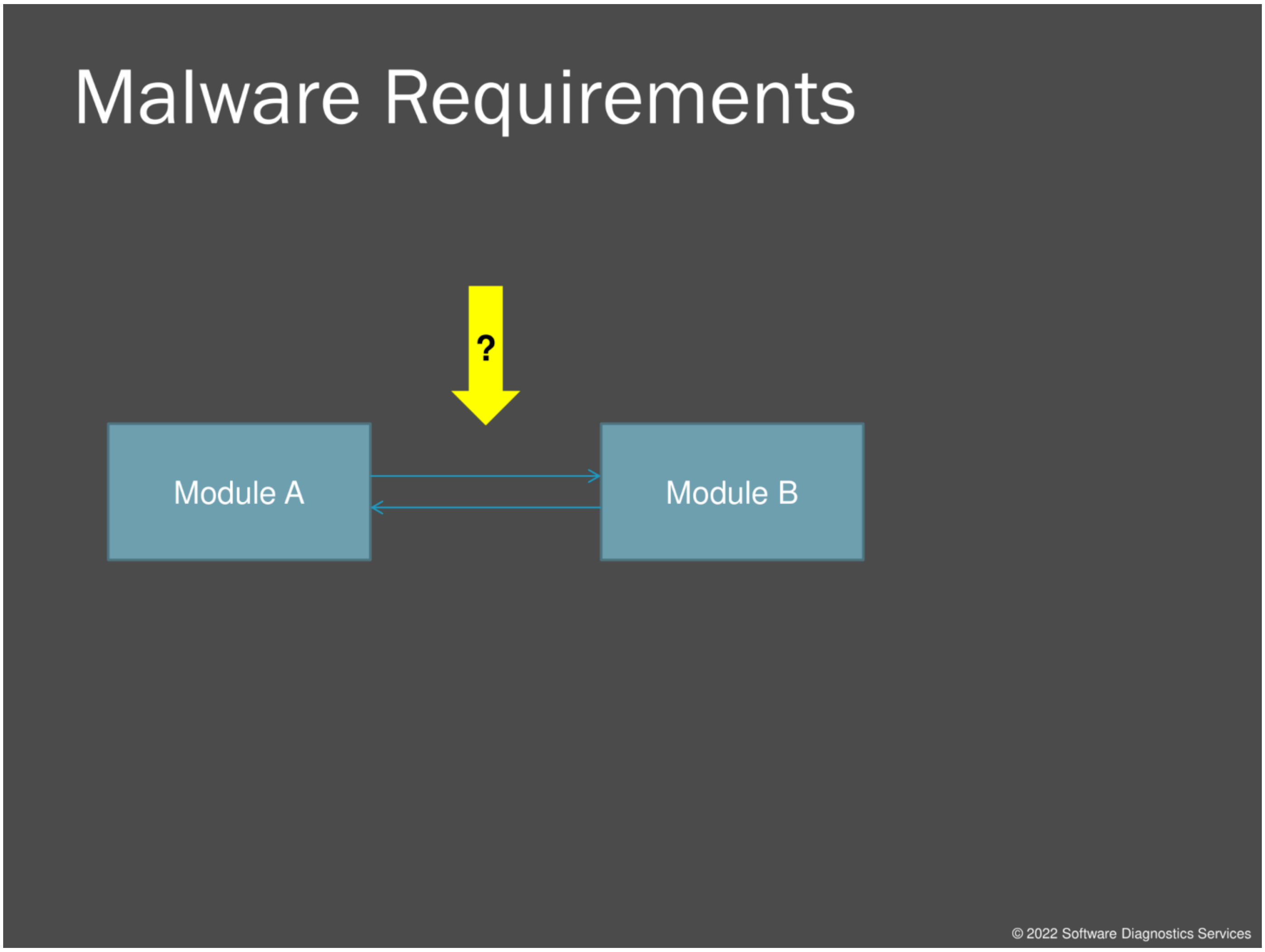
```
0:000> s-sa 012e4000 L6600
012e40fd  "!This program cannot be run in D"
012e411d  "OS mode."
012e4188  "Rich"
012e4290  "UPX0"
012e42b8  "UPX1"
012e42e0  ".rsrc"
012e448b  "3.08"
012e4490  "UPX!"
012e449c  "9T5"
012e44d1  "vqx"
[...]
012e84b8  "%BoxW"
012e84c2  "ActiveWindowas"
[...]
012e9197  "o:\Work\AWMA\M2\ReleaseN"
[...]
012e94da  "ommand"
012e94f1  "IsDe"
012e94f9  "buggerP"
[...]
```

11. We can even write this embedded binary to some folder and try to unpack it (012e40b0 is an address of the "MZ" signature) and then later load an unpacked version as a crash dump for further analysis:

```
0:000> .writemem c:\AWMA-Dumps\module.bin 012e40b0 L6600
Writing 6600 bytes.............
```

12. Close the log file:

```
0:000> .logclose
Closing open log file C:\AWMA-Dumps\M2.log
```



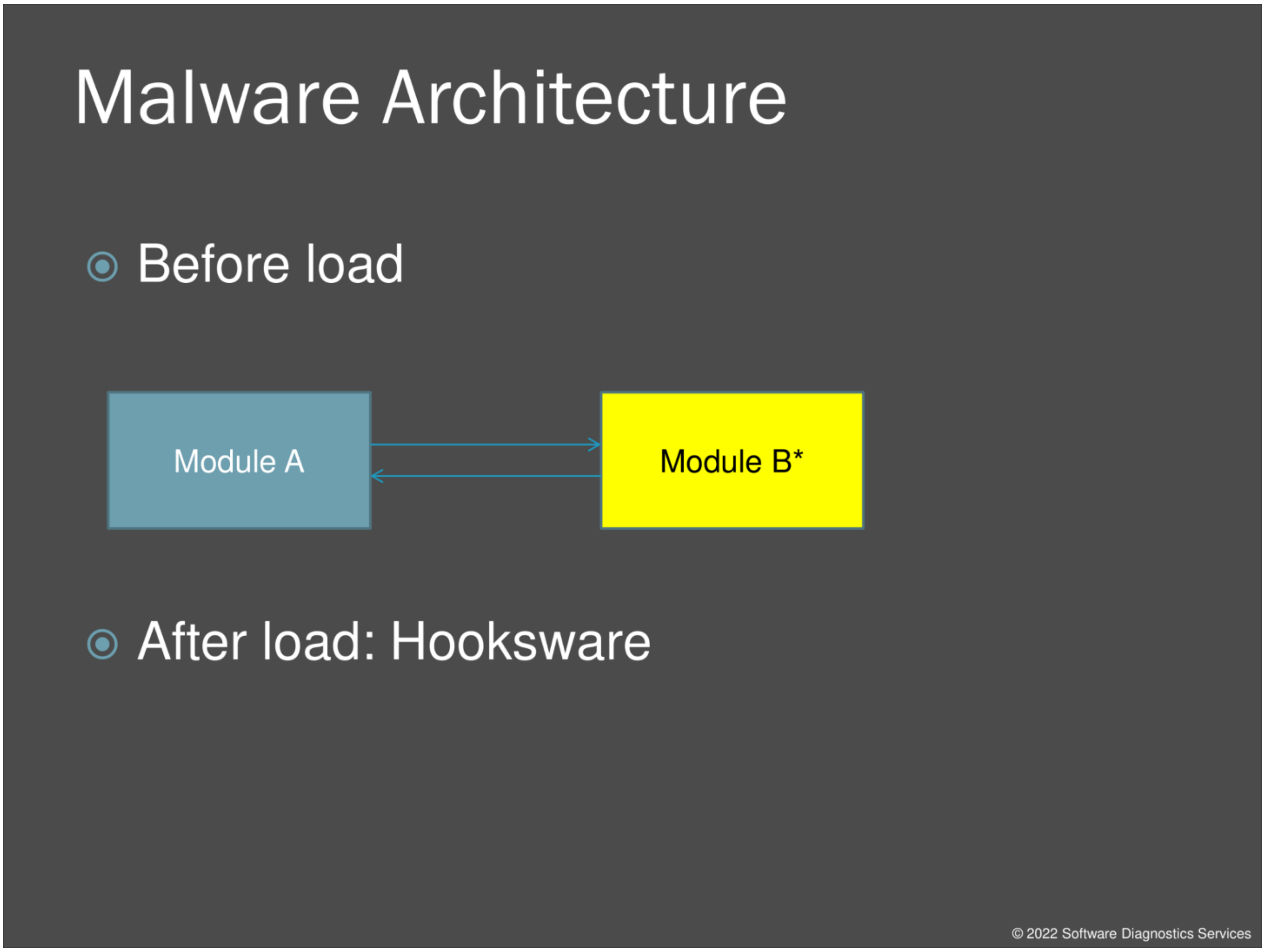
Malware Requirements

?

Module A

Module B

For malware to do something malicious, it needs to be executed. So its basic requirement is to be loaded into memory and get the attention of a CPU.



Malware Architecture

- Before load

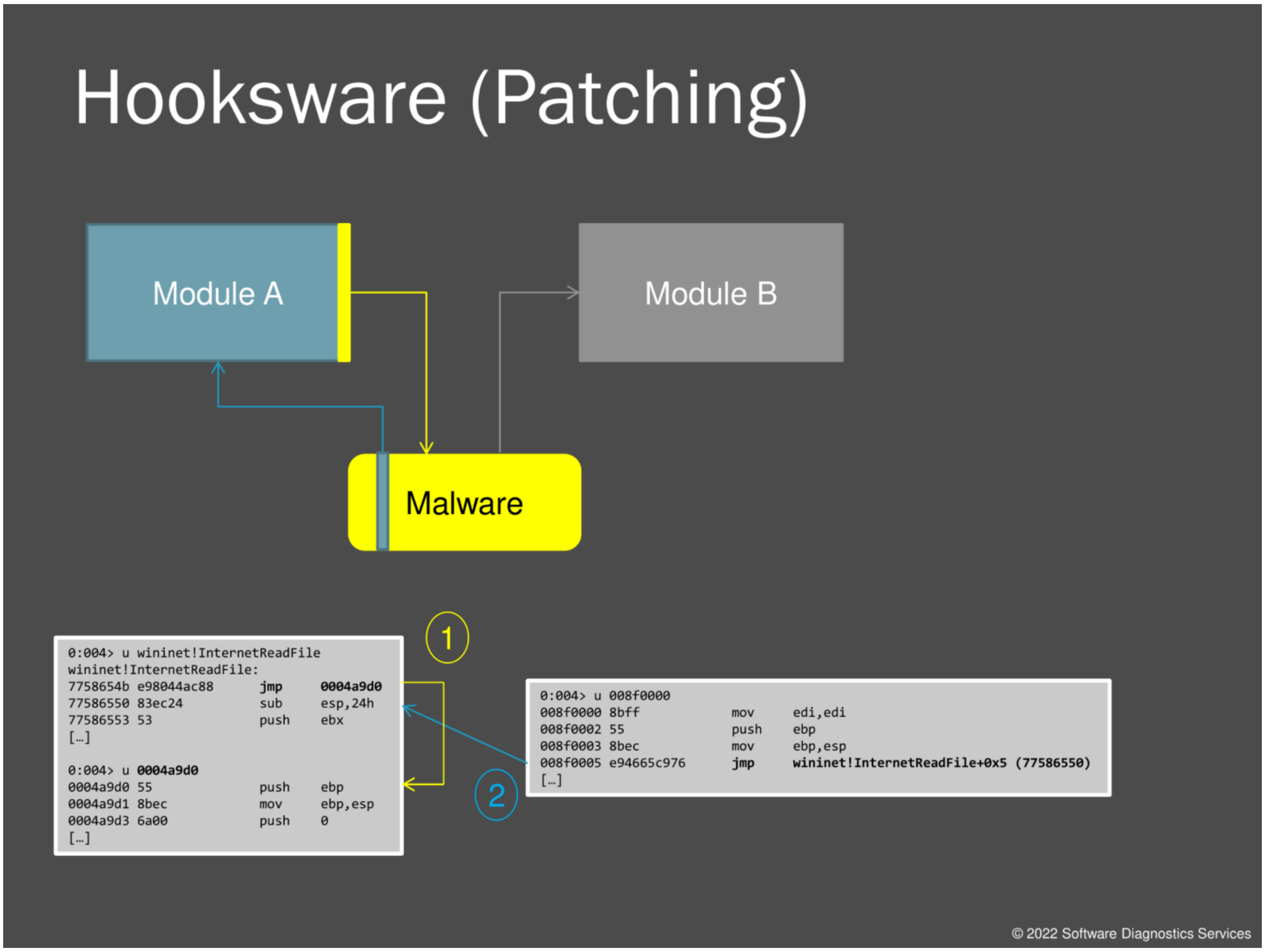
Module A

Module B*

- After load: Hookware

© 2022 Software Diagnostics Services

Such requirements can be implemented by replacing modules with fake ones or somehow modifying existing modules before they are loaded into memory. Another way is when genuine malware modules are loaded, and they modify existing modules and structures in memory resulting in execution being redirected to them, the so-called hooksware method that combines various approaches such as windows hooks, patching, and DLL injection by remote thread execution.

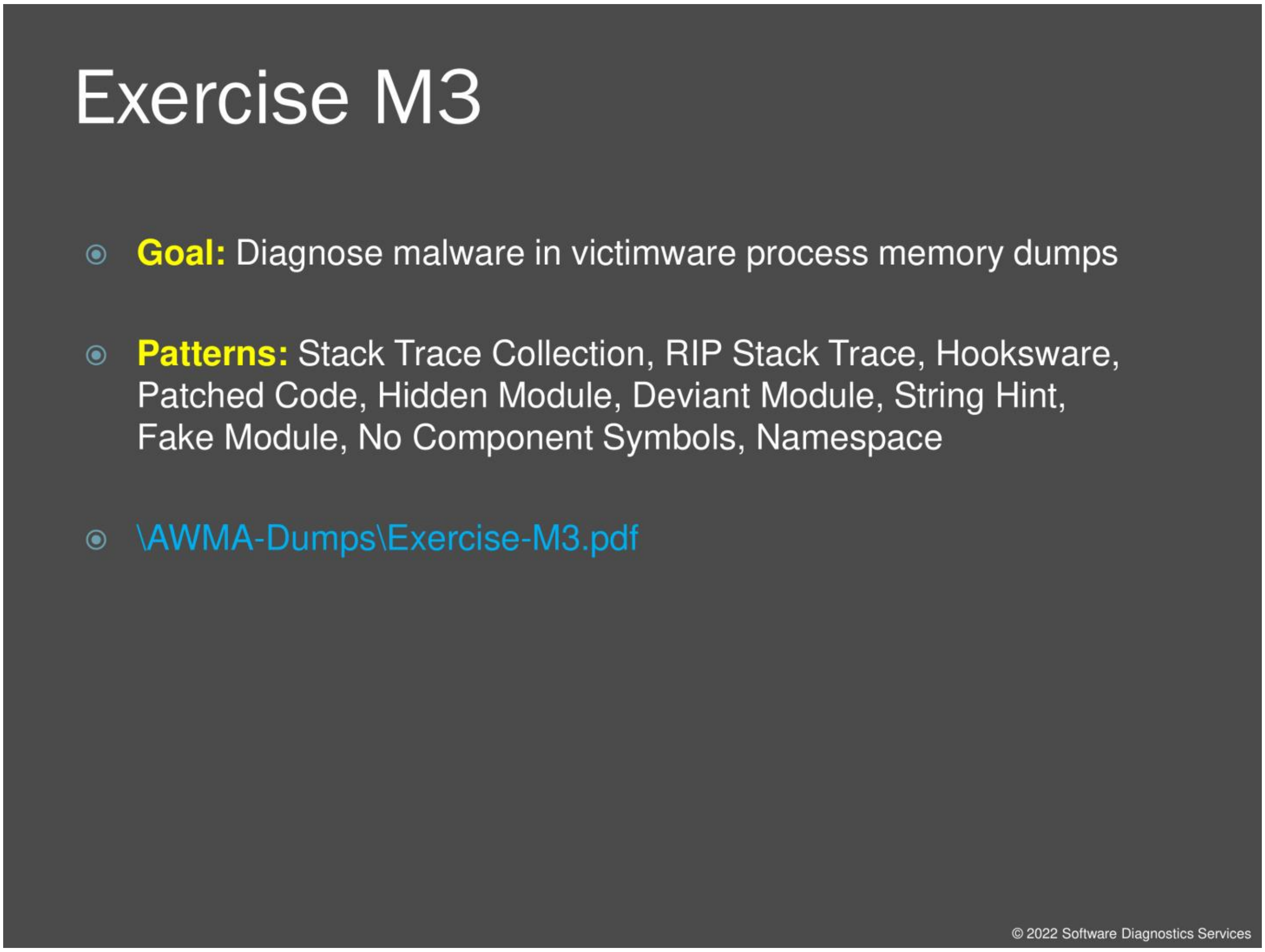


Hooksware (Patching)

```
0:004> u wininet!InternetReadFile
wininet!InternetReadFile:
7758654b e98044ac88      jmp     0004a9d0
77586550 83ec24          sub     esp,24h
77586553 53              push    ebx
[…]

0:004> u 0004a9d0
0004a9d0 55              push    ebp
0004a9d1 8bec            mov     ebp,esp
0004a9d3 6a00            push    0
[…]
```

```
0:004> u 008f0000
008f0000 8bff            mov     edi,edi
008f0002 55              push    ebp
008f0003 8bec            mov     ebp,esp
008f0005 e94665c976      jmp     wininet!InternetReadFile+0x5 (77586550)
[…]
```

Here we cover only code patching and delegate to a free Debugging TV session for the DLL Injection case study (See Frame 0x20 episode on www.debugging.tv). In the forthcoming exercise, you see these patching effects in action. Basically, the initial code in a function is saved and replaced by a jump to another code region, and after malicious activity, execution is returned back to the previous code after executing its saved portion.



Exercise M3

- **Goal:** Diagnose malware in victimware process memory dumps
- **Patterns:** Stack Trace Collection, RIP Stack Trace, Hooksware, Patched Code, Hidden Module, Deviant Module, String Hint, Fake Module, No Component Symbols, Namespace
- \AWMA-Dumps\Exercise-M3.pdf

Now we analyze a real malware crash dump with many malware analysis patterns.

Exercise M3

Goal: Diagnose malware in victimware[1] process memory dumps.

Patterns: Stack Trace Collection, RIP Stack Trace, Hooksware, Patched Code, Hidden Module, Deviant Module, String Hint, Fake Module, No Component Symbols, Namespace.

1. Launch WinDbg Preview.

2. Open \AWMA-Dumps\Processes\iexplore.exe.5564.dmp.

3. We get the dump file loaded:

```
Microsoft (R) Windows Debugger Version 10.0.25136.1001 X86
Copyright (c) Microsoft Corporation. All rights reserved.


Loading Dump File [C:\AWMA-Dumps\Processes\iexplore.exe.5564.dmp]
User Mini Dump File with Full Memory: Only application data is available


************* Path validation summary **************
Response                         Time (ms)     Location
Deferred                                       srv*
Symbol search path is: srv*
Executable search path is:
Windows Server 2008/Windows Vista Version 6002 (Service Pack 2) MP (2 procs) Free x86
compatible
Product: WinNt, suite: SingleUserTS Personal
Machine Name:
Debug session time: Sun Sep 26 09:19:07.000 2010 (UTC + 1:00)
System Uptime: 0 days 18:41:40.127
Process Uptime: 0 days 0:00:48.000
................................................................
.........................................................
Loading unloaded module list
..
This dump file has an exception of interest stored in it.
The stored exception information can be accessed via .ecxr.
(15bc.650): Unknown exception - code c0000374 (first/second chance not available)
For analysis of this file, run !analyze -v
eax=00000000 ebx=00000000 ecx=00000400 edx=00000000 esi=026e0000 edi=000015bc
eip=77815e74 esp=02c9cb1c ebp=02c9cba0 iopl=0         nv up ei pl nz na po nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00040202
ntdll!KiFastSystemCallRet:
77815e74 c3              ret
```

Note the message about a stored exception.

4. Open a log file:

```
0:004> .logopen C:\AWMA-Dumps\M3.log
Opened log file 'C:\AWMA-Dumps\M3.log'
```

[1] Victimware vs. Malware was first introduced here: https://www.patterndiagnostics.com/files/Victimware.pdf

5. We first try to use **!analyze -v** command:

```
0:004> !analyze -v
*******************************************************************************
*                                                                             *
*                        Exception Analysis                                   *
*                                                                             *
*******************************************************************************

*************************************************************************
***                                                                   ***
***                                                                   ***
***    Either you specified an unqualified symbol, or your debugger   ***
***    doesn't have full symbol information.  Unqualified symbol      ***
***    resolution is turned off by default. Please either specify a   ***
***    fully qualified symbol module!symbolname, or enable resolution ***
***    of unqualified symbols by typing ".symopt- 100". Note that     ***
***    enabling unqualified symbol resolution with network symbol     ***
***    server shares in the symbol path may cause the debugger to     ***
***    appear to hang for long periods of time when an incorrect      ***
***    symbol name is typed or the network symbol server is down.     ***
***                                                                   ***
***    For some commands to work properly, your symbol path           ***
***    must point to .pdb files that have full type information.      ***
***                                                                   ***
***    Certain .pdb files (such as the public OS symbols) do not      ***
***    contain the required information.  Contact the group that      ***
***    provided you with these symbols if you need this command to    ***
***    work.                                                          ***
***                                                                   ***
***    Type referenced: kernel32!pNlsUserInfo                         ***
***                                                                   ***
*************************************************************************
*************************************************************************
***                                                                   ***
***                                                                   ***
***    Either you specified an unqualified symbol, or your debugger   ***
***    doesn't have full symbol information.  Unqualified symbol      ***
***    resolution is turned off by default. Please either specify a   ***
***    fully qualified symbol module!symbolname, or enable resolution ***
***    of unqualified symbols by typing ".symopt- 100". Note that     ***
***    enabling unqualified symbol resolution with network symbol     ***
***    server shares in the symbol path may cause the debugger to     ***
***    appear to hang for long periods of time when an incorrect      ***
***    symbol name is typed or the network symbol server is down.     ***
***                                                                   ***
***    For some commands to work properly, your symbol path           ***
***    must point to .pdb files that have full type information.      ***
***                                                                   ***
***    Certain .pdb files (such as the public OS symbols) do not      ***
***    contain the required information.  Contact the group that      ***
***    provided you with these symbols if you need this command to    ***
***    work.                                                          ***
***                                                                   ***
***    Type referenced: kernel32!pNlsUserInfo                         ***
***                                                                   ***
*************************************************************************

KEY_VALUES_STRING: 1
```

```
    Key  : Analysis.CPU.mSec
    Value: 10765

    Key  : Analysis.DebugAnalysisManager
    Value: Create

    Key  : Analysis.Elapsed.mSec
    Value: 31965

    Key  : Analysis.Init.CPU.mSec
    Value: 1952

    Key  : Analysis.Init.Elapsed.mSec
    Value: 1609855

    Key  : Analysis.Memory.CommitPeak.Mb
    Value: 132

    Key  : Timeline.OS.Boot.DeltaSec
    Value: 67300

    Key  : Timeline.Process.Start.DeltaSec
    Value: 48

    Key  : WER.OS.Branch
    Value: lh_sp2rtm

    Key  : WER.OS.Timestamp
    Value: 2009-04-10T18:30:00Z

    Key  : WER.OS.Version
    Value: 6.0.6002.18005

    Key  : WER.Process.Version
    Value: 8.0.6001.18943


FILE_IN_CAB:  iexplore.exe.5564.dmp

NTGLOBALFLAG:  400

PROCESS_BAM_CURRENT_THROTTLED: 0

PROCESS_BAM_PREVIOUS_THROTTLED: 0

APPLICATION_VERIFIER_FLAGS:  0

CONTEXT:  (.ecxr)
eax=02c9d01c ebx=00000000 ecx=7fffffff edx=00000000 esi=00290000 edi=04f1ffe0
eip=7785faf8 esp=02c9d00c ebp=02c9d084 iopl=0         nv up ei pl zr na pe nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00040246
ntdll!RtlReportCriticalFailure+0x5b:
7785faf8 eb1c            jmp     ntdll!RtlReportCriticalFailure+0x6f (7785fb16)
Resetting default scope

EXCEPTION_RECORD:  (.exr -1)
ExceptionAddress: 7785faf8 (ntdll!RtlReportCriticalFailure+0x0000005b)
   ExceptionCode: c0000374
  ExceptionFlags: 00000001
NumberParameters: 1
```

```
   Parameter[0]: 7787c040

PROCESS_NAME:  iexplore.exe

ERROR_CODE: (NTSTATUS) 0xc0000374 - A heap has been corrupted.

EXCEPTION_CODE_STR:  c0000374

EXCEPTION_PARAMETER1:  7787c040

ADDITIONAL_DEBUG_TEXT:  Followup set based on attribute [Heap_Error_Type] from Frame:[0] on
thread:[PSEUDO_THREAD] ; Followup set based on attribute [Is_ChosenCrashFollowupThread] from
Frame:[0] on thread:[PSEUDO_THREAD]

FAULTING_THREAD:  ffffffff

STACK_TEXT:
00000000 00000000 urlmon!ReleaseBindInfo+0x0


SYMBOL_NAME:  urlmon!ReleaseBindInfo+0

MODULE_NAME: urlmon

IMAGE_NAME:  urlmon.dll

STACK_COMMAND:  .ecxr ; kb ; !heap ; ** Pseudo Context ** ManagedPseudo ** Value: ffffffff ** ;
kb

FAILURE_BUCKET_ID:
HEAP_CORRUPTION_ACTIONABLE_EntryCorruption_c0000374_urlmon.dll!ReleaseBindInfo

OS_VERSION:  6.0.6002.18005

BUILDLAB_STR:  lh_sp2rtm

OSPLATFORM_TYPE:  x86

OSNAME:  Windows Vista

IMAGE_VERSION:  8.0.6001.18943

FAILURE_ID_HASH:  {cfc9f375-dd8e-ac69-2897-b6988ca80919}

Followup:     MachineOwner
---------
```

We see heap corruption diagnostics. And the stack trace confirms that:

```
0:004> k
 # ChildEBP RetAddr
00 02c9cb18 77815620     ntdll!KiFastSystemCallRet
01 02c9cb1c 77843c62     ntdll!ZwWaitForSingleObject+0xc
02 02c9cba0 77843d4b     ntdll!RtlReportExceptionEx+0x14b
03 02c9cbe0 7785fa87     ntdll!RtlReportException+0x3c
04 02c9cbf4 7785fb0d     ntdll!RtlpTerminateFailureFilter+0x14
05 02c9cc00 777b9bdc     ntdll!RtlReportCriticalFailure+0x6b
06 02c9cc14 777b4067     ntdll!_EH4_CallFilterFunc+0x12
07 02c9cc3c 77815f79     ntdll!_except_handler4+0x8e
08 02c9cc60 77815f4b     ntdll!ExecuteHandler2+0x26
```

```
09 02c9cd10 77815dd7     ntdll!ExecuteHandler+0x24
0a 02c9cd10 7785faf8     ntdll!KiUserExceptionDispatcher+0xf
0b 02c9d084 77860704     ntdll!RtlReportCriticalFailure+0x5b
0c 02c9d094 778607f2     ntdll!RtlpReportHeapFailure+0x21
0d 02c9d0c8 7782b1a5     ntdll!RtlpLogHeapFailure+0xa1
0e 02c9d110 7781730a     ntdll!RtlpCoalesceFreeBlocks+0x4b9
0f 02c9d208 77817545     ntdll!RtlpFreeHeap+0x1e2
10 02c9d224 76277e4b     ntdll!RtlFreeHeap+0x14e
11 02c9d26c 760f7277     kernel32!GlobalFree+0x47
12 02c9d280 76594a1f     ole32!ReleaseStgMedium+0x124 [d:\longhorn\com\ole32\ole232\base\api.cpp @ 964]
13 02c9d294 765f7feb     urlmon!ReleaseBindInfo+0x4c
14 02c9d2a4 765b9a87     urlmon!CINet::ReleaseCNetObjects+0x3d
15 02c9d2bc 765b93f0     urlmon!CINetHttp::OnWininetRequestHandleClosing+0x60
16 02c9d2d0 77582078     urlmon!CINet::CINetCallback+0x2de
17 02c9d418 77588f5d     wininet!InternetIndicateStatus+0xfc
18 02c9d448 7758937a     wininet!HANDLE_OBJECT::~HANDLE_OBJECT+0xc9
19 02c9d464 7758916b     wininet!INTERNET_CONNECT_HANDLE_OBJECT::~INTERNET_CONNECT_HANDLE_OBJECT+0x209
1a 02c9d470 77588d5e     wininet!HTTP_REQUEST_HANDLE_OBJECT::`scalar deleting destructor'+0xd
1b 02c9d480 77584e72     wininet!HANDLE_OBJECT::Dereference+0x22
1c 02c9d48c 77589419     wininet!DereferenceObject+0x21
1d 02c9d4b4 77589114     wininet!_InternetCloseHandle+0x9d
1e 02c9d4d4 0004aaaf     wininet!InternetCloseHandle+0x11e
WARNING: Frame IP not in any known module. Following frames may be wrong.
1f 02c9d4e0 765a5d25     0x4aaaf
20 02c9d4fc 765a5c1b     urlmon!CINet::TerminateRequest+0x82
21 02c9d50c 765a5a3c     urlmon!CINet::MyTerminate+0x7b
22 02c9d51c 765a5998     urlmon!CINetProtImpl::Terminate+0x13
23 02c9d538 765a5b92     urlmon!CINetEmbdFilter::Terminate+0x17
24 02c9d548 765b9bc1     urlmon!CINet::Terminate+0x23
25 02c9d55c 765979f2     urlmon!CINetHttp::Terminate+0x48
26 02c9d574 7659766b     urlmon!COInetProt::Terminate+0x1d
27 02c9d598 765979c0     urlmon!CTransaction::Terminate+0x12d
28 02c9d5b8 76597a2d     urlmon!CBinding::ReportResult+0x92
29 02c9d5d0 76596609     urlmon!COInetProt::ReportResult+0x1a
2a 02c9d5f8 76596322     urlmon!CTransaction::DispatchReport+0x1d9
2b 02c9d624 7659653e     urlmon!CTransaction::DispatchPacket+0x31
2c 02c9d644 765a504b     urlmon!CTransaction::OnINetCallback+0x92
2d 02c9d65c 7741fd72     urlmon!TransactionWndProc+0x28
2e 02c9d688 7741fe4a     user32!InternalCallWinProc+0x23
2f 02c9d700 7742018d     user32!UserCallWinProcCheckWow+0x14b
30 02c9d764 7742022b     user32!DispatchMessageWorker+0x322
31 02c9d774 7094c1d5     user32!DispatchMessageW+0xf
32 02c9f87c 708f337e     ieframe!CTabWindow::_TabWindowThreadProc+0x54c
33 02c9f934 7647426d     ieframe!LCIETab_ThreadProc+0x2c1
34 02c9f944 7627d0e9     iertutil!CIsoScope::RegisterThread+0xab
35 02c9f950 777f19bb     kernel32!BaseThreadInitThunk+0xe
36 02c9f990 777f198e     ntdll!__RtlUserThreadStart+0x23
37 02c9f9a8 00000000     ntdll!_RtlUserThreadStart+0x1b
```

The usual impulse here is to enable a full page heap (where memory is allocated at the end of pages with the next page invalid to catch buffer overruns) and collect a new dump. We also do it but now analyze the dump a bit further.

6. Let's check stack traces from all process threads:

```
0:004> ~*kL

   0  Id: 15bc.12c4 Suspend: 1 Teb: 7ffdf000 Unfrozen
 # ChildEBP RetAddr
00 001df4d8 77815610     ntdll!KiFastSystemCallRet
01 001df4dc 7627a5d7     ntdll!ZwWaitForMultipleObjects+0xc
02 001df578 77420f8d     kernel32!WaitForMultipleObjectsEx+0x11d
03 001df5cc 7647334a     user32!RealMsgWaitForMultipleObjectsEx+0x13c
04 001df61c 76474942     iertutil!IsoDispatchMessageToArtifacts+0x22c
05 001df63c 708c416a     iertutil!IsoManagerThreadZero_WindowsPump+0x52
06 001df68c 00ff12e3     ieframe!LCIEStartAsTabProcess+0x25f
```

```
07 001df7d8 00ff147a     iexplore!wWinMain+0x368
08 001df86c 7627d0e9     iexplore!_initterm_e+0x1b1
09 001df878 777f19bb     kernel32!BaseThreadInitThunk+0xe
0a 001df8b8 777f198e     ntdll!__RtlUserThreadStart+0x23
0b 001df8d0 00000000     ntdll!_RtlUserThreadStart+0x1b

   1  Id: 15bc.17a8 Suspend: 1 Teb: 7ffde000 Unfrozen
 # ChildEBP RetAddr
00 0258f6d8 77815610     ntdll!KiFastSystemCallRet
01 0258f6dc 777f2934     ntdll!ZwWaitForMultipleObjects+0xc
02 0258f870 7627d0e9     ntdll!TppWaiterpThread+0x328
03 0258f87c 777f19bb     kernel32!BaseThreadInitThunk+0xe
04 0258f8bc 777f198e     ntdll!__RtlUserThreadStart+0x23
05 0258f8d4 00000000     ntdll!_RtlUserThreadStart+0x1b

   2  Id: 15bc.1148 Suspend: 1 Teb: 7ffdc000 Unfrozen
 # ChildEBP RetAddr
00 02a2ed3c 77815610     ntdll!KiFastSystemCallRet
01 02a2ed40 7627a5d7     ntdll!ZwWaitForMultipleObjects+0xc
02 02a2eddc 7627a6f0     kernel32!WaitForMultipleObjectsEx+0x11d
03 02a2edf8 7646f08c     kernel32!WaitForMultipleObjects+0x18
04 02a2fe24 76474819     iertutil!CForeignProcessToCurrentProcessMessaging::_vThreadProc+0xa1
05 02a2fe2c 7627d0e9     iertutil!CForeignProcessToCurrentProcessMessaging::_sThreadProc+0xd
06 02a2fe38 777f19bb     kernel32!BaseThreadInitThunk+0xe
07 02a2fe78 777f198e     ntdll!__RtlUserThreadStart+0x23
08 02a2fe90 00000000     ntdll!_RtlUserThreadStart+0x1b

   3  Id: 15bc.9e8 Suspend: 1 Teb: 7ffdb000 Unfrozen
 # ChildEBP RetAddr
00 028ef9a8 77815610     ntdll!KiFastSystemCallRet
01 028ef9ac 7627a5d7     ntdll!ZwWaitForMultipleObjects+0xc
02 028efa48 77420f8d     kernel32!WaitForMultipleObjectsEx+0x11d
03 028efa9c 7647334a     user32!RealMsgWaitForMultipleObjectsEx+0x13c
04 028efaec 764748b6     iertutil!IsoDispatchMessageToArtifacts+0x22c
05 028efb0c 7627d0e9     iertutil!IsoManagerThreadNonzero_WindowsPump+0x59
06 028efb18 777f19bb     kernel32!BaseThreadInitThunk+0xe
07 028efb58 777f198e     ntdll!__RtlUserThreadStart+0x23
08 028efb70 00000000     ntdll!_RtlUserThreadStart+0x1b

#  4  Id: 15bc.650 Suspend: 0 Teb: 7ffda000 Unfrozen
 # ChildEBP RetAddr
00 02c9cb18 77815620     ntdll!KiFastSystemCallRet
01 02c9cb1c 77843c62     ntdll!ZwWaitForSingleObject+0xc
02 02c9cba0 77843d4b     ntdll!RtlReportExceptionEx+0x14b
03 02c9cbe0 7785fa87     ntdll!RtlReportException+0x3c
04 02c9cbf4 7785fb0d     ntdll!RtlpTerminateFailureFilter+0x14
05 02c9cc00 777b9bdc     ntdll!RtlReportCriticalFailure+0x6b
06 02c9cc14 777b4067     ntdll!_EH4_CallFilterFunc+0x12
07 02c9cc3c 77815f79     ntdll!_except_handler4+0x8e
08 02c9cc60 77815f4b     ntdll!ExecuteHandler2+0x26
09 02c9cd10 77815dd7     ntdll!ExecuteHandler+0x24
0a 02c9cd10 7785faf8     ntdll!KiUserExceptionDispatcher+0xf
0b 02c9d084 77860704     ntdll!RtlReportCriticalFailure+0x5b
0c 02c9d094 778607f2     ntdll!RtlpReportHeapFailure+0x21
0d 02c9d0c8 7782b1a5     ntdll!RtlpLogHeapFailure+0xa1
0e 02c9d110 7781730a     ntdll!RtlpCoalesceFreeBlocks+0x4b9
0f 02c9d208 77817545     ntdll!RtlpFreeHeap+0x1e2
10 02c9d224 76277e4b     ntdll!RtlFreeHeap+0x14e
11 02c9d26c 760f7277     kernel32!GlobalFree+0x47
12 02c9d280 76594a1f     ole32!ReleaseStgMedium+0x124
13 02c9d294 765f7feb     urlmon!ReleaseBindInfo+0x4c
14 02c9d2a4 765b9a87     urlmon!CINet::ReleaseCNetObjects+0x3d
15 02c9d2bc 765b93f0     urlmon!CINetHttp::OnWininetRequestHandleClosing+0x60
16 02c9d2d0 77582078     urlmon!CINet::CINetCallback+0x2de
17 02c9d418 77588f5d     wininet!InternetIndicateStatus+0xfc
18 02c9d448 7758937a     wininet!HANDLE_OBJECT::~HANDLE_OBJECT+0xc9
19 02c9d464 7758916b     wininet!INTERNET_CONNECT_HANDLE_OBJECT::~INTERNET_CONNECT_HANDLE_OBJECT+0x209
```

```
1a 02c9d470 77588d5e     wininet!HTTP_REQUEST_HANDLE_OBJECT::`scalar deleting destructor'+0xd
1b 02c9d480 77584e72     wininet!HANDLE_OBJECT::Dereference+0x22
1c 02c9d48c 77589419     wininet!DereferenceObject+0x21
1d 02c9d4b4 77589114     wininet!_InternetCloseHandle+0x9d
1e 02c9d4d4 0004aaaf     wininet!InternetCloseHandle+0x11e
WARNING: Frame IP not in any known module. Following frames may be wrong.
1f 02c9d4e0 765a5d25     0x4aaaf
20 02c9d4fc 765a5c1b     urlmon!CINet::TerminateRequest+0x82
21 02c9d50c 765a5a3c     urlmon!CINet::MyTerminate+0x7b
22 02c9d51c 765a5998     urlmon!CINetProtImpl::Terminate+0x13
23 02c9d538 765a5b92     urlmon!CINetEmbdFilter::Terminate+0x17
24 02c9d548 765b9bc1     urlmon!CINet::Terminate+0x23
25 02c9d55c 765979f2     urlmon!CINetHttp::Terminate+0x48
26 02c9d574 7659766b     urlmon!COInetProt::Terminate+0x1d
27 02c9d598 765979c0     urlmon!CTransaction::Terminate+0x12d
28 02c9d5b8 76597a2d     urlmon!CBinding::ReportResult+0x92
29 02c9d5d0 76596609     urlmon!COInetProt::ReportResult+0x1a
2a 02c9d5f8 76596322     urlmon!CTransaction::DispatchReport+0x1d9
2b 02c9d624 7659653e     urlmon!CTransaction::DispatchPacket+0x31
2c 02c9d644 765a504b     urlmon!CTransaction::OnINetCallback+0x92
2d 02c9d65c 7741fd72     urlmon!TransactionWndProc+0x28
2e 02c9d688 7741fe4a     user32!InternalCallWinProc+0x23
2f 02c9d700 7742018d     user32!UserCallWinProcCheckWow+0x14b
30 02c9d764 7742022b     user32!DispatchMessageWorker+0x322
31 02c9d774 7094c1d5     user32!DispatchMessageW+0xf
32 02c9f87c 708f337e     ieframe!CTabWindow::_TabWindowThreadProc+0x54c
33 02c9f934 7647426d     ieframe!LCIETab_ThreadProc+0x2c1
34 02c9f944 7627d0e9     iertutil!CIsoScope::RegisterThread+0xab
35 02c9f950 777f19bb     kernel32!BaseThreadInitThunk+0xe
36 02c9f990 777f198e     ntdll!__RtlUserThreadStart+0x23
37 02c9f9a8 00000000     ntdll!_RtlUserThreadStart+0x1b

   5  Id: 15bc.efc Suspend: 1 Teb: 7ffd9000 Unfrozen
 # ChildEBP RetAddr
00 02e8fa48 77815610     ntdll!KiFastSystemCallRet
01 02e8fa4c 7627a5d7     ntdll!ZwWaitForMultipleObjects+0xc
02 02e8fae8 7627a6f0     kernel32!WaitForMultipleObjectsEx+0x11d
03 02e8fb04 275c55c0     kernel32!WaitForMultipleObjects+0x18
WARNING: Stack unwind information not available. Following frames may be wrong.
04 02e8fc4c 777f4123     msidcrl40!CreatePassportAuthUIContext+0x2ab30
05 02e8fc88 777f3e23     ntdll!RtlpTpTimerCallback+0x62
06 02e8fcac 777f2fcf     ntdll!TppTimerpExecuteCallback+0x14d
07 02e8fddc 7627d0e9     ntdll!TppWorkerThread+0x545
08 02e8fde8 777f19bb     kernel32!BaseThreadInitThunk+0xe
09 02e8fe28 777f198e     ntdll!__RtlUserThreadStart+0x23
0a 02e8fe40 00000000     ntdll!_RtlUserThreadStart+0x1b

   6  Id: 15bc.10ec Suspend: 1 Teb: 7ffd8000 Unfrozen
 # ChildEBP RetAddr
00 0409fd70 77814780     ntdll!KiFastSystemCallRet
01 0409fd74 76279990     ntdll!NtDelayExecution+0xc
02 0409fddc 76231c6c     kernel32!SleepEx+0x62
03 0409fdec 76123f1d     kernel32!Sleep+0xf
04 0409fdf8 7613eb46     ole32!CROIDTable::WorkerThreadLoop+0x14
05 0409fe14 761257ab     ole32!CRpcThread::WorkerLoop+0x26
06 0409fe24 7627d0e9     ole32!CRpcThreadCache::RpcWorkerThreadEntry+0x16
07 0409fe30 777f19bb     kernel32!BaseThreadInitThunk+0xe
08 0409fe70 777f198e     ntdll!__RtlUserThreadStart+0x23
09 0409fe88 00000000     ntdll!_RtlUserThreadStart+0x1b

   7  Id: 15bc.1500 Suspend: 1 Teb: 7ffd6000 Unfrozen
 # ChildEBP RetAddr
00 03f0fb68 778150b0     ntdll!KiFastSystemCallRet
01 03f0fb6c 7627d11e     ntdll!NtRemoveIoCompletion+0xc
02 03f0fb98 75ec03c8     kernel32!GetQueuedCompletionStatus+0x29
03 03f0fbd4 75ec04fd     rpcrt4!COMMON_ProcessCalls+0xb5
04 03f0fc44 75ec011c     rpcrt4!LOADABLE_TRANSPORT::ProcessIOEvents+0x138
```

```
05 03f0fc4c 75ec00e3     rpcrt4!ProcessIOEventsWrapper+0xd
06 03f0fc70 75ec0166     rpcrt4!BaseCachedThreadRoutine+0x5c
07 03f0fc7c 7627d0e9     rpcrt4!ThreadStartRoutine+0x1e
08 03f0fc88 777f19bb     kernel32!BaseThreadInitThunk+0xe
09 03f0fcc8 777f198e     ntdll!__RtlUserThreadStart+0x23
0a 03f0fce0 00000000     ntdll!_RtlUserThreadStart+0x1b

   8  Id: 15bc.1364 Suspend: 1 Teb: 7ffd5000 Unfrozen
 # ChildEBP RetAddr
00 0474f5f8 77815620     ntdll!KiFastSystemCallRet
01 0474f5fc 75471aa6     ntdll!ZwWaitForSingleObject+0xc
02 0474f63c 7547179d     mswsock!SockWaitForSingleObject+0x19f
03 0474f728 77381693     mswsock!WSPSelect+0x38c
04 0474f7a8 7757e9a9     ws2_32!select+0x494
05 0474fb00 7759deab     wininet!ICAsyncThread::SelectThread+0x242
06 0474fb08 7627d0e9     wininet!ICAsyncThread::SelectThreadWrapper+0xd
07 0474fb14 777f19bb     kernel32!BaseThreadInitThunk+0xe
08 0474fb54 777f198e     ntdll!__RtlUserThreadStart+0x23
09 0474fb6c 00000000     ntdll!_RtlUserThreadStart+0x1b

   9  Id: 15bc.1224 Suspend: 1 Teb: 7ffaf000 Unfrozen
 # ChildEBP RetAddr
00 051ff8a8 778157b0     ntdll!KiFastSystemCallRet
01 051ff8ac 777f2eb0     ntdll!NtWaitForWorkViaWorkerFactory+0xc
02 051ff9dc 7627d0e9     ntdll!TppWorkerThread+0x1f6
03 051ff9e8 777f19bb     kernel32!BaseThreadInitThunk+0xe
04 051ffa28 777f198e     ntdll!__RtlUserThreadStart+0x23
05 051ffa40 00000000     ntdll!_RtlUserThreadStart+0x1b

  10  Id: 15bc.990 Suspend: 1 Teb: 7ffad000 Unfrozen
 # ChildEBP RetAddr
00 04dbf860 778150b0     ntdll!KiFastSystemCallRet
01 04dbf864 754764f1     ntdll!NtRemoveIoCompletion+0xc
02 04dbf89c 7627d0e9     mswsock!SockAsyncThread+0x69
03 04dbf8a8 777f19bb     kernel32!BaseThreadInitThunk+0xe
04 04dbf8e8 777f198e     ntdll!__RtlUserThreadStart+0x23
05 04dbf900 00000000     ntdll!_RtlUserThreadStart+0x1b

  11  Id: 15bc.fa4 Suspend: 1 Teb: 7ffac000 Unfrozen
 # ChildEBP RetAddr
00 0568fe78 77815620     ntdll!KiFastSystemCallRet
01 0568fe7c 76279884     ntdll!ZwWaitForSingleObject+0xc
02 0568feec 762797f2     kernel32!WaitForSingleObjectEx+0xbe
03 0568ff00 6ca4a731     kernel32!WaitForSingleObject+0x12
04 0568ff24 6c9b0778     mshtml!CDwnTaskExec::ThreadExec+0x23c
05 0568ff2c 6c9b083b     mshtml!CExecFT::ThreadProc+0x39
06 0568ff38 7627d0e9     mshtml!CExecFT::StaticThreadProc+0xe
07 0568ff44 777f19bb     kernel32!BaseThreadInitThunk+0xe
08 0568ff84 777f198e     ntdll!__RtlUserThreadStart+0x23
09 0568ff9c 00000000     ntdll!_RtlUserThreadStart+0x1b

  12  Id: 15bc.d10 Suspend: 1 Teb: 7ffaa000 Unfrozen
 # ChildEBP RetAddr
00 06e1fca0 77815620     ntdll!KiFastSystemCallRet
01 06e1fca4 76279884     ntdll!ZwWaitForSingleObject+0xc
02 06e1fd14 762797f2     kernel32!WaitForSingleObjectEx+0xbe
03 06e1fd28 6ca4a731     kernel32!WaitForSingleObject+0x12
04 06e1fd4c 6c9b0778     mshtml!CDwnTaskExec::ThreadExec+0x23c
05 06e1fd54 6c9b083b     mshtml!CExecFT::ThreadProc+0x39
06 06e1fd60 7627d0e9     mshtml!CExecFT::StaticThreadProc+0xe
07 06e1fd6c 777f19bb     kernel32!BaseThreadInitThunk+0xe
08 06e1fdac 777f198e     ntdll!__RtlUserThreadStart+0x23
09 06e1fdc4 00000000     ntdll!_RtlUserThreadStart+0x1b

  13  Id: 15bc.294 Suspend: 1 Teb: 7ffa9000 Unfrozen
 # ChildEBP RetAddr
00 06f1f6dc 77815610     ntdll!KiFastSystemCallRet
```

```
01 06f1f6e0 7627a5d7     ntdll!ZwWaitForMultipleObjects+0xc
02 06f1f77c 7627a6f0     kernel32!WaitForMultipleObjectsEx+0x11d
03 06f1f798 275b4879     kernel32!WaitForMultipleObjects+0x18
WARNING: Stack unwind information not available. Following frames may be wrong.
04 06f1fabc 275b4a58     msidcrl40!CreatePassportAuthUIContext+0x19de9
05 06f1fae4 275c9655     msidcrl40!CreatePassportAuthUIContext+0x19fc8
06 06f1fb1c 275c96fa     msidcrl40!CreatePassportAuthUIContext+0x2ebc5
07 06f1fb30 777f19bb     msidcrl40!CreatePassportAuthUIContext+0x2ec6a
08 06f1fb70 777f198e     ntdll!__RtlUserThreadStart+0x23
09 06f1fb88 00000000     ntdll!_RtlUserThreadStart+0x1b

  14  Id: 15bc.ebc Suspend: 1 Teb: 7ffa8000 Unfrozen
 # ChildEBP RetAddr
00 0775f5fc 77815610     ntdll!KiFastSystemCallRet
01 0775f600 7627a5d7     ntdll!ZwWaitForMultipleObjects+0xc
02 0775f69c 7627a6f0     kernel32!WaitForMultipleObjectsEx+0x11d
03 0775f6b8 275b4879     kernel32!WaitForMultipleObjects+0x18
WARNING: Stack unwind information not available. Following frames may be wrong.
04 0775f9dc 275b4a58     msidcrl40!CreatePassportAuthUIContext+0x19de9
05 0775fa04 275c9655     msidcrl40!CreatePassportAuthUIContext+0x19fc8
06 0775fa3c 275c96fa     msidcrl40!CreatePassportAuthUIContext+0x2ebc5
07 0775fa50 777f19bb     msidcrl40!CreatePassportAuthUIContext+0x2ec6a
08 0775fa90 777f198e     ntdll!__RtlUserThreadStart+0x23
09 0775faa8 00000000     ntdll!_RtlUserThreadStart+0x1b

  15  Id: 15bc.99c Suspend: 1 Teb: 7ffa6000 Unfrozen
 # ChildEBP RetAddr
00 0501faf4 778157b0     ntdll!KiFastSystemCallRet
01 0501faf8 777f2eb0     ntdll!NtWaitForWorkViaWorkerFactory+0xc
02 0501fc28 7627d0e9     ntdll!TppWorkerThread+0x1f6
03 0501fc34 777f19bb     kernel32!BaseThreadInitThunk+0xe
04 0501fc74 777f198e     ntdll!__RtlUserThreadStart+0x23
05 0501fc8c 00000000     ntdll!_RtlUserThreadStart+0x1b

  16  Id: 15bc.1128 Suspend: 1 Teb: 7ffa5000 Unfrozen
 # ChildEBP RetAddr
00 0785f748 77815620     ntdll!KiFastSystemCallRet
01 0785f74c 76279884     ntdll!ZwWaitForSingleObject+0xc
02 0785f7bc 762797f2     kernel32!WaitForSingleObjectEx+0xbe
03 0785f7d0 6ca4a731     kernel32!WaitForSingleObject+0x12
04 0785f7f0 6c9b0778     mshtml!CDwnTaskExec::ThreadExec+0x23c
05 0785f7f8 6c9b083b     mshtml!CExecFT::ThreadProc+0x39
06 0785f804 7627d0e9     mshtml!CExecFT::StaticThreadProc+0xe
07 0785f810 777f19bb     kernel32!BaseThreadInitThunk+0xe
08 0785f850 777f198e     ntdll!__RtlUserThreadStart+0x23
09 0785f868 00000000     ntdll!_RtlUserThreadStart+0x1b

  17  Id: 15bc.b44 Suspend: 1 Teb: 7ffa1000 Unfrozen
 # ChildEBP RetAddr
00 0868fc78 77815620     ntdll!KiFastSystemCallRet
01 0868fc7c 76279884     ntdll!ZwWaitForSingleObject+0xc
02 0868fcec 762797f2     kernel32!WaitForSingleObjectEx+0xbe
03 0868fd00 6cbe8fed     kernel32!WaitForSingleObject+0x12
04 0868fd24 6c9b0778     mshtml!CTimerMan::ThreadExec+0x90
05 0868fd2c 6c9b083b     mshtml!CExecFT::ThreadProc+0x39
06 0868fd38 7627d0e9     mshtml!CExecFT::StaticThreadProc+0xe
07 0868fd44 777f19bb     kernel32!BaseThreadInitThunk+0xe
08 0868fd84 777f198e     ntdll!__RtlUserThreadStart+0x23
09 0868fd9c 00000000     ntdll!_RtlUserThreadStart+0x1b

  18  Id: 15bc.4d0 Suspend: 1 Teb: 7ffa0000 Unfrozen
 # ChildEBP RetAddr
00 0b99fbbc 7741feef     ntdll!KiFastSystemCallRet
01 0b99fbc0 77418af3     user32!NtUserGetMessage+0xc
02 0b99fbe4 7450145c     user32!GetMessageA+0x8a
03 0b99fc1c 7627d0e9     winmm!mciwindow+0x102
04 0b99fc28 777f19bb     kernel32!BaseThreadInitThunk+0xe
```

```
05 0b99fc68 777f198e     ntdll!__RtlUserThreadStart+0x23
06 0b99fc80 00000000     ntdll!_RtlUserThreadStart+0x1b

  19  Id: 15bc.e10 Suspend: 1 Teb: 7ff9f000 Unfrozen
 # ChildEBP RetAddr
00 0bc7fa20 77815610     ntdll!KiFastSystemCallRet
01 0bc7fa24 7627a5d7     ntdll!ZwWaitForMultipleObjects+0xc
02 0bc7fac0 742d4f1d     kernel32!WaitForMultipleObjectsEx+0x11d
03 0bc7faf8 742d7e96     wdmaud!CWorker::_ThreadProc+0x5e
04 0bc7fb04 7627d0e9     wdmaud!CWorker::_StaticThreadProc+0x18
05 0bc7fb10 777f19bb     kernel32!BaseThreadInitThunk+0xe
06 0bc7fb50 777f198e     ntdll!__RtlUserThreadStart+0x23
07 0bc7fb68 00000000     ntdll!_RtlUserThreadStart+0x1b

  20  Id: 15bc.15b0 Suspend: 1 Teb: 7ffa4000 Unfrozen
 # ChildEBP RetAddr
00 0b04fc00 77815610     ntdll!KiFastSystemCallRet
01 0b04fc04 7627a5d7     ntdll!ZwWaitForMultipleObjects+0xc
02 0b04fca0 77420f8d     kernel32!WaitForMultipleObjectsEx+0x11d
03 0b04fcf4 77417f5a     user32!RealMsgWaitForMultipleObjectsEx+0x13c
04 0b04fd10 745974b2     user32!MsgWaitForMultipleObjects+0x1f
05 0b04fd5c 7627d0e9     GdiPlus!BackgroundThreadProc+0x59
06 0b04fd68 777f19bb     kernel32!BaseThreadInitThunk+0xe
07 0b04fda8 777f198e     ntdll!__RtlUserThreadStart+0x23
08 0b04fdc0 00000000     ntdll!_RtlUserThreadStart+0x1b

  21  Id: 15bc.15a8 Suspend: 1 Teb: 7ffdd000 Unfrozen
 # ChildEBP RetAddr
00 0bb7fb08 778150b0     ntdll!KiFastSystemCallRet
01 0bb7fb0c 7627d11e     ntdll!NtRemoveIoCompletion+0xc
02 0bb7fb38 75ec03c8     kernel32!GetQueuedCompletionStatus+0x29
03 0bb7fb74 75ec04fd     rpcrt4!COMMON_ProcessCalls+0xb5
04 0bb7fbe4 75ec011c     rpcrt4!LOADABLE_TRANSPORT::ProcessIOEvents+0x138
05 0bb7fbec 75ec00e3     rpcrt4!ProcessIOEventsWrapper+0xd
06 0bb7fc14 75ec0166     rpcrt4!BaseCachedThreadRoutine+0x5c
07 0bb7fc20 7627d0e9     rpcrt4!ThreadStartRoutine+0x1e
08 0bb7fc2c 777f19bb     kernel32!BaseThreadInitThunk+0xe
09 0bb7fc6c 777f198e     ntdll!__RtlUserThreadStart+0x23
0a 0bb7fc84 00000000     ntdll!_RtlUserThreadStart+0x1b
```

The only problem thread we see is #4 with exception processing code after detected heap corruption. What we also see is a raw instruction pointer **0x4aaaf** in the stack trace. This can often be seen in managed .NET execution environment with its JIT-compiled .NET code. However, there is no presence of .NET CLR modules such as *mscorwks.dll*, *clr.dll,* or *coreclr.dll* in the stack trace.

7. Let's look at this RIP address closely by doing backwards disassembly:

```
0:004> ub 0x4aaaf
0004aa97 740c            je      0004aaa5
0004aa99 8b4508          mov     eax,dword ptr [ebp+8]
0004aa9c 50              push    eax
0004aa9d e82eedffff      call    000497d0
0004aaa2 83c404          add     esp,4
0004aaa5 8b4d08          mov     ecx,dword ptr [ebp+8]
0004aaa8 51              push    ecx
0004aaa9 ff1580aa0500    call    dword ptr ds:[5AA80h]
```

Note that there is an indirect call through another address **5AA80**:

```
0:004> db 5AA80
0005aa80  00 00 93 00 00 00 8f 00-00 00 27 00 00 00 90 00  ..........'.....
0005aa90  00 00 25 00 00 00 dc 01-4d 6f 7a 69 6c 6c 61 2f  ..%.....Mozilla/
```

```
0005aaa0  34 2e 30 20 28 63 6f 6d-70 61 74 69 62 6c 65 3b  4.0 (compatible;
0005aab0  20 4d 53 49 45 20 38 2e-30 3b 20 57 69 6e 64 6f   MSIE 8.0; Windo
0005aac0  77 73 20 4e 54 20 36 2e-30 3b 20 54 72 69 64 65  ws NT 6.0; Tride
0005aad0  6e 74 2f 34 2e 30 3b 20-4d 61 74 68 50 6c 61 79  nt/4.0; MathPlay
0005aae0  65 72 20 32 2e 31 30 64-3b 20 53 4c 43 43 31 3b  er 2.10d; SLCC1;
0005aaf0  20 2e 4e 45 54 20 43 4c-52 20 32 2e 30 2e 35 30   .NET CLR 2.0.50
```

```
0:004> dps 5AA80
0005aa80  00930000
0005aa84  008f0000
0005aa88  00270000
0005aa8c  00900000
0005aa90  00250000
0005aa94  01dc0000
0005aa98  697a6f4d
0005aa9c  2f616c6c
0005aaa0  20302e34
0005aaa4  6d6f6328
0005aaa8  69746170
0005aaac  3b656c62
0005aab0  49534d20
0005aab4  2e382045
0005aab8  57203b30
0005aabc  6f646e69
0005aac0  4e207377
0005aac4  2e362054
0005aac8  54203b30
0005aacc  65646972
0005aad0  342f746e
0005aad4  203b302e
0005aad8  6874614d
0005aadc  79616c50
0005aae0  32207265
0005aae4  6430312e
0005aae8  4c53203b
0005aaec  3b314343
0005aaf0  454e2e20
0005aaf4  4c432054
0005aaf8  2e322052
0005aafc  30352e30
```

```
0:004> u 00930000
00930000 8bff            mov     edi,edi
00930002 55              push    ebp
00930003 8bec            mov     ebp,esp
00930005 e98390c576      jmp     wininet!InternetCloseHandle+0x5 (7758908d)
0093000a 0000            add     byte ptr [eax],al
0093000c 0000            add     byte ptr [eax],al
0093000e 0000            add     byte ptr [eax],al
00930010 0000            add     byte ptr [eax],al
```

Let’s check all other addresses from **dps** command output before ASCII data:

```
0:004> u 008f0000
008f0000 8bff            mov     edi,edi
008f0002 55              push    ebp
008f0003 8bec            mov     ebp,esp
008f0005 e94665c976      jmp     wininet!InternetReadFile+0x5 (77586550)
008f000a 0000            add     byte ptr [eax],al
008f000c 0000            add     byte ptr [eax],al
```

```
008f000e 0000            add     byte ptr [eax],al
008f0010 0000            add     byte ptr [eax],al

0:004> u 00270000
00270000 8bff            mov     edi,edi
00270002 55              push    ebp
00270003 8bec            mov     ebp,esp
00270005 e905a73877      jmp     wininet!HttpSendRequestExA+0x5 (775fa70f)
0027000a 0000            add     byte ptr [eax],al
0027000c 0000            add     byte ptr [eax],al
0027000e 0000            add     byte ptr [eax],al
00270010 0000            add     byte ptr [eax],al

0:004> u 00900000
00900000 8bff            mov     edi,edi
00900002 55              push    ebp
00900003 8bec            mov     ebp,esp
00900005 e97c33ca76      jmp     wininet!InternetReadFileExA+0x5 (775a3386)
0090000a 0000            add     byte ptr [eax],al
0090000c 0000            add     byte ptr [eax],al
0090000e 0000            add     byte ptr [eax],al
00900010 0000            add     byte ptr [eax],al

0:004> u 00250000
00250000 8bff            mov     edi,edi
00250002 55              push    ebp
00250003 8bec            mov     ebp,esp
00250005 e984ee3477      jmp     wininet!HttpSendRequestA+0x5 (7759ee8e)
0025000a 0000            add     byte ptr [eax],al
0025000c 0000            add     byte ptr [eax],al
0025000e 0000            add     byte ptr [eax],al
00250010 0000            add     byte ptr [eax],al
```

All these code jumps look like a return to the original hooked function code. Let's check the first instructions in all these functions:

```
0:004> u wininet!InternetCloseHandle
wininet!InternetCloseHandle:
77589088 e9031aac88      jmp     0004aa90
7758908d 51              push    ecx
7758908e 51              push    ecx
7758908f 53              push    ebx
77589090 56              push    esi
77589091 57              push    edi
77589092 33db            xor     ebx,ebx
77589094 33ff            xor     edi,edi

0:004> u wininet!InternetReadFile
wininet!InternetReadFile:
7758654b e98044ac88      jmp     0004a9d0
77586550 83ec24          sub     esp,24h
77586553 53              push    ebx
77586554 56              push    esi
77586555 57              push    edi
77586556 33ff            xor     edi,edi
77586558 393db8116277    cmp     dword ptr [wininet!GlobalDataInitialized (776211b8)],edi
7758655e 897df4          mov     dword ptr [ebp-0Ch],edi
```

```
0:004> u wininet!HttpSendRequestExA
wininet!HttpSendRequestExA:
775fa70a e9f1faa488      jmp     0004a200
775fa70f 53              push    ebx
775fa710 56              push    esi
775fa711 57              push    edi
775fa712 33db            xor     ebx,ebx
775fa714 33c9            xor     ecx,ecx
775fa716 33d2            xor     edx,edx
775fa718 33f6            xor     esi,esi

0:004> u wininet!InternetReadFileExA
wininet!InternetReadFileExA:
775a3381 e97a76aa88      jmp     0004aa00
775a3386 83ec20          sub     esp,20h
775a3389 53              push    ebx
775a338a 33db            xor     ebx,ebx
775a338c 391db8116277    cmp     dword ptr [wininet!GlobalDataInitialized (776211b8)],ebx
775a3392 56              push    esi
775a3393 57              push    edi
775a3394 895dfc          mov     dword ptr [ebp-4],ebx

0:004> u wininet!HttpSendRequestA
wininet!HttpSendRequestA:
7759ee89 e952b2aa88      jmp     0004a0e0
7759ee8e 6a10            push    10h
7759ee90 6a00            push    0
7759ee92 ff7518          push    dword ptr [ebp+18h]
7759ee95 ff7514          push    dword ptr [ebp+14h]
7759ee98 ff7510          push    dword ptr [ebp+10h]
7759ee9b ff750c          push    dword ptr [ebp+0Ch]
7759ee9e ff7508          push    dword ptr [ebp+8]
```

Let’s check the address attribute:

```
0:004> !address 0x4aaaf


Mapping file section regions...
Mapping module regions...
Mapping PEB regions...
Mapping TEB and stack regions...
*** Failure in mapping Heap (80004005: ExtRemoteTyped::Field: unable to retrieve field
'BaseAddress' at ffffffff99654a5f)
Mapping page heap regions...
Mapping other regions...
Mapping stack trace database regions...
Mapping activation context regions...

Usage:                  <unknown>
Base Address:           00040000
End Address:            0005d000
Region Size:            0001d000 ( 116.000 kB)
State:                  00001000          MEM_COMMIT
Protect:                00000040          PAGE_EXECUTE_READWRITE
Type:                   00020000          MEM_PRIVATE
Allocation Base:        00040000
Allocation Protect:     00000040          PAGE_EXECUTE_READWRITE
```

```
Content source: 1 (target), length: 12551
```

We see that the region is also writable compared to normal code:

```
0:004> !address 775fa70a


Usage:                  Image
Base Address:           77571000
End Address:            77621000
Region Size:            000b0000
State:                  00001000 MEM_COMMIT
Protect:                00000020 PAGE_EXECUTE_READ
Type:                   01000000 MEM_IMAGE
Allocation Base:        77570000
Allocation Protect:     00000080 PAGE_EXECUTE_WRITECOPY
Image Path:             C:\Windows\System32\wininet.dll
Module Name:            wininet
Loaded Image Name:      wininet.dll
Mapped Image Name:
More info:              lmv m wininet
More info:              !lmi wininet
More info:              ln 0x775fa70a
More info:              !dh 0x77570000
```

8. Now we check if the base address contains any module information:

```
0:004> dc 00040000
00040000  00905a4d 00000003 00000004 0000ffff  MZ..............
00040010  000000b8 00000000 00000040 00000000  ........@.......
00040020  00000000 00000000 00000000 00000000  ................
00040030  00000000 00000000 00000000 000000d8  ................
00040040  0eba1f0e cd09b400 4c01b821 685421cd  ........!..L.!Th
00040050  70207369 72676f72 63206d61 6f6e6e61  is program canno
00040060  65622074 6e757220 206e6920 20534f44  t be run in DOS
00040070  65646f6d 0a0d0d2e 00000024 00000000  mode....$.......

0:004> !dh 00040000

File Type: EXECUTABLE IMAGE
FILE HEADER VALUES
     14C machine (i386)
       4 number of sections
4C9E36D3 time date stamp Sat Sep 25 18:52:19 2010

       0 file pointer to symbol table
       0 number of symbols
      E0 size of optional header
     102 characteristics
            Executable
            32 bit word machine

OPTIONAL HEADER VALUES
     10B magic #
    9.00 linker version
   12200 size of code
    7000 size of initialized data
       0 size of uninitialized data
    D5F0 address of entry point
    1000 base of code
         ----- new -----
00400000 image base
    1000 section alignment
     200 file alignment
       2 subsystem (Windows GUI)
    5.00 operating system version
    0.00 image version
    5.00 subsystem version
   1D000 size of image
     400 size of headers
       0 checksum
00100000 size of stack reserve
00001000 size of stack commit
00100000 size of heap reserve
00001000 size of heap commit
    8540  DLL characteristics
            Dynamic base
            NX compatible
            No structured exception handler
            Terminal server aware
       0 [       0] address [size] of Export Directory
       0 [       0] address [size] of Import Directory
       0 [       0] address [size] of Resource Directory
       0 [       0] address [size] of Exception Directory
       0 [       0] address [size] of Security Directory
   1C000 [     3F0] address [size] of Base Relocation Directory
       0 [       0] address [size] of Debug Directory
       0 [       0] address [size] of Description Directory
       0 [       0] address [size] of Special Directory
       0 [       0] address [size] of Thread Storage Directory
       0 [       0] address [size] of Load Configuration Directory
       0 [       0] address [size] of Bound Import Directory
       0 [       0] address [size] of Import Address Table Directory
       0 [       0] address [size] of Delay Import Directory
       0 [       0] address [size] of COR20 Header Directory
       0 [       0] address [size] of Reserved Directory


SECTION HEADER #1
   .text name
   1203B virtual size
    1000 virtual address
   12200 size of raw data
     400 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
60000020 flags
         Code
         (no align specified)
         Execute Read

SECTION HEADER #2
  .rdata name
     7D0 virtual size
   14000 virtual address
     800 size of raw data
   12600 file pointer to raw data
```

```
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
40000040 flags
         Initialized Data
         (no align specified)
         Read Only

SECTION HEADER #3
   .data name
    6008 virtual size
   15000 virtual address
    4000 size of raw data
   12E00 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
C0000040 flags
         Initialized Data
         (no align specified)
         Read Write

SECTION HEADER #4
  .reloc name
     5F0 virtual size
   1C000 virtual address
     600 size of raw data
   16E00 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
42000040 flags
         Initialized Data
         Discardable
         (no align specified)
         Read Only
```

We see the module doesn’t have any import tables.

9. We now check the module range for any string hints:

```
0:004> s-sa 00040000 0005d000
0004004d  "!This program cannot be run in D"
0004006d  "OS mode."
00040081  "3y@"
000400b8  "Rich"
000401d0  ".text"
000401f7  "`.rdata"
0004021f  "@.data"
00040248  ".reloc"
[...]
00054000  "HELLO"
00054008  "%s:%s"
00054010  "READY"
00054018  "GET /stat?uptime=%d&downlink=%d&"
00054038  "uplink=%d&id=%s&statpass=%s&comm"
```

```
00054058  "ent=%s HTTP/1.0"
000540ac  "%s%s%s"
000540d8  "ftp://%s:%s@%s:%d"
000540fc  "Accept-Encoding:"
00054118  "Accept-Encoding:"
00054130  "0123456789ABCDEF"
00054144  "://"
00054160  "POST %s HTTP/1.0"
00054172  "Host: %s"
0005417c  "User-Agent: %s"
0005418c  "Accept: text/html"
0005419f  "Connection: Close"
000541b2  "Content-Type: application/x-www-"
000541d2  "form-urlencoded"
000541e3  "Content-Length: %d"
000541fc  "id="
00054208  "POST %s HTTP/1.1"
0005421a  "Host: %s"
00054224  "User-Agent: %s"
00054234  "Accept: text/html"
00054247  "Connection: Close"
0005425a  "Content-Type: application/x-www-"
0005427a  "form-urlencoded"
0005428b  "Content-Length: %d"
000542a4  "id=%s&base="
000542b8  "id=%s&brw=%d&type=%d&data="
000542d8  "POST %s HTTP/1.1"
000542ea  "Host: %s"
000542f4  "User-Agent: %s"
00054304  "Accept: text/html"
00054317  "Connection: Close"
0005432a  "Content-Type: application/x-www-"
0005434a  "form-urlencoded"
0005435b  "Content-Length: %d"
00054378  "id=%s&os=%s&plist="
00054390  "POST %s HTTP/1.1"
000543a2  "Host: %s"
000543ac  "User-Agent: %s"
000543bc  "Accept: text/html"
000543cf  "Connection: Close"
000543e2  "Content-Type: application/x-www-"
00054402  "form-urlencoded"
00054413  "Content-Length: %d"
00054430  "id=%s&data=%s"
00054440  "POST %s HTTP/1.1"
00054452  "Host: %s"
0005445c  "User-Agent: %s"
0005446c  "Accept: text/html"
0005447f  "Connection: Close"
00054492  "Content-Type: application/x-www-"
000544b2  "form-urlencoded"
000544c3  "Content-Length: %d"
000544e0  "GET %s HTTP/1.0"
000544f1  "Host: %s"
000544fb  "User-Agent: %s"
0005450b  "Connection: close"
00054528  "POST /get/scr.html HTTP/1.0"
00054545  "Host: %s"
0005454f  "User-Agent: %s"
0005455f  "Connection: close"
```

```
00054572  "Content-Length: %d"
00054586  "Content-Type: multipart/form-dat"
000545a6  "a; boundary=--------------------"
000545c6  "-------%d"
000545d4  "-----------------------------%d"
000545f8  "%sContent-Disposition: form-data"
00054618  "; name="id""
00054630  "%sContent-Disposition: form-data"
00054650  "; name="screen"; filename="%d""
00054670  "Content-Type: application/octet-"
00054690  "stream"
000546a0  "%s(%d) : %s"
000546ac  "%s failed with error %d: %s"
000546c8  "%02X"
000546d8  "BlackwoodPRO"
000546e8  "FinamDirect"
000546f4  "GrayBox"
000546fc  "MbtPRO"
00054704  "Laser"
0005470c  "LightSpeed"
00054718  "LTGroup"
00054720  "Mbt"
00054724  "ScotTrader"
00054730  "SaxoTrader"
00054740  "Program:   %s"
0005474f  "Username:  %s"
0005475e  "Password:  %s"
0005476d  "AccountNO: %s"
0005477c  "Server:    %s"
00054790  "%s %s"
0005479c  "PROCESSOR_IDENTIFIER"
[...]
0005a8e0  "glebk"
0005aa98  "Mozilla/4.0 (compatible; MSIE 8."
0005aab8  "0; Windows NT 6.0; Trident/4.0; "
0005aad8  "MathPlayer 2.10d; SLCC1; .NET CL"
0005aaf8  "R 2.0.50727; Media Center PC 5.0"
0005ab18  "; .NET CLR 3.5.30729; .NET CLR 3"
0005ab38  ".0.30729)"
[...]

0:004> s-su 00040000 0005d000
[...]
00055004  "\chkntfs.exe"
00055020  "\chkntfs.dat"
[...]
00058e20  "kernel32.dll"
00058e3c  "user32.dll"
00058e54  "ws2_32.dll"
00058e6c  "ntdll.dll"
00058e80  "wininet.dll"
00058e98  "nspr4.dll"
00058eac  "ssl3.dll"
0005a4e0  "C:\Users\dima\AppData\Roaming\ch"
0005a520  "kntfs.dat"
[...]
```

We find some references to the fake *chkntfs.exe* here and the list of modules needed for this malware. Also, “gleb” is a Russian name, but it could be just a coincidence.

10. Let’s now check if there are any Hidden Modules not shown in the loaded module list by using the **.imgscan** command that searches for MZ/PE signatures:

```
0:004> .imgscan
MZ at 00040000, prot 00000040, type 00020000 - size 1d000
MZ at 00fa0000, prot 00000002, type 00040000 - size 2000
MZ at 00ff0000, prot 00000002, type 01000000 - size 9c000
  Name: iexplore.exe
MZ at 044b0000, prot 00000002, type 00040000 - size 2000
MZ at 08f50000, prot 00000002, type 01000000 - size 335000
  Name: igdumd32.dll
MZ at 0a390000, prot 00000002, type 00040000 - size 191000
MZ at 10000000, prot 00000004, type 00020000 - size 5000
  Name: screens_dll.dll
MZ at 16080000, prot 00000002, type 01000000 - size 25000
  Name: mdnsNSP.dll
MZ at 27500000, prot 00000002, type 01000000 - size 11a000
  Name: msidcrl40.dll
MZ at 29500000, prot 00000002, type 01000000 - size 67000
  Name: IDBHO.DLL
MZ at 633d0000, prot 00000002, type 01000000 - size 4f000
  Name: rpbrowserrecordplugin.dll
MZ at 634b0000, prot 00000002, type 01000000 - size 1d000
  Name: rpchromebrowserrecordhelper.dll
MZ at 68f80000, prot 00000002, type 01000000 - size 5e3000
  Name: Flash.ocx
MZ at 6a2b0000, prot 00000002, type 01000000 - size 45b000
  Name: agcore.dll
MZ at 6bfb0000, prot 00000002, type 01000000 - size d8000
  Name: NPCTRL.dll
MZ at 6c8c0000, prot 00000002, type 01000000 - size 6a000
  Name: VBSCRIPT.dll
MZ at 6c9a0000, prot 00000002, type 01000000 - size 5b0000
  Name: MSHTML.dll
MZ at 6d150000, prot 00000002, type 01000000 - size 39000
  Name: dxtrans.dll
MZ at 6d1d0000, prot 00000002, type 01000000 - size b4000
  Name: JSCRIPT.dll
MZ at 6d2c0000, prot 00000002, type 01000000 - size a000
  Name: DDRAWEX.DLL
MZ at 6d3e0000, prot 00000002, type 01000000 - size e000
  Name: PNGFILTER.DLL
MZ at 6d440000, prot 00000002, type 01000000 - size c000
  Name: jp2ssv.dll
MZ at 6dbf0000, prot 00000002, type 01000000 - size 33000
  Name: IEShims.dll
MZ at 6e080000, prot 00000002, type 01000000 - size 29000
  Name: msls31.dll
MZ at 6e100000, prot 00000002, type 01000000 - size 40000
  Name: SWEEPRX.dll
MZ at 6e150000, prot 00000002, type 01000000 - size 2f000
  Name: iepeers.DLL
MZ at 6e520000, prot 00000002, type 01000000 - size b000
  Name: msimtf.dll
MZ at 6e550000, prot 00000002, type 01000000 - size c000
  Name: ImgUtil.dll
MZ at 6e8a0000, prot 00000002, type 01000000 - size 1b000
  Name: CRYPTNET.dll
MZ at 6e960000, prot 00000002, type 01000000 - size 26000
```

```
  Name: DSSENH.dll
MZ at 6ea00000, prot 00000002, type 01000000 - size 30000
  Name: MLANG.dll
MZ at 6f320000, prot 00000002, type 01000000 - size 6000
  Name: SensApi.dll
MZ at 6f340000, prot 00000002, type 01000000 - size 31000
  Name: TAPI32.dll
MZ at 6f3c0000, prot 00000002, type 01000000 - size 14000
  Name: rasman.dll
MZ at 6f3e0000, prot 00000002, type 01000000 - size 4a000
  Name: RASAPI32.dll
MZ at 6f840000, prot 00000002, type 01000000 - size 70000
  Name: DSOUND.dll
MZ at 6f8d0000, prot 00000002, type 01000000 - size 136000
  Name: MSXML3.dll
MZ at 6fa40000, prot 00000002, type 01000000 - size c000
  Name: rtutils.dll
MZ at 70320000, prot 00000002, type 01000000 - size 3e000
  Name: pdh.dll
MZ at 70620000, prot 00000002, type 01000000 - size e5000
  Name: DDRAW.dll
MZ at 70820000, prot 00000002, type 01000000 - size a94000
  Name: IEFRAME.dll
MZ at 71a70000, prot 00000002, type 01000000 - size 62000
  Name: mscms.dll
MZ at 71bb0000, prot 00000002, type 01000000 - size 12000
  Name: PNRPNSP.dll
MZ at 723c0000, prot 00000002, type 01000000 - size 53000
  Name: SWEEPRX.dll
MZ at 72430000, prot 00000002, type 01000000 - size 42000
  Name: WINSPOOL.DRV
MZ at 72ff0000, prot 00000002, type 01000000 - size 6000
  Name: rasadhlp.dll
MZ at 73320000, prot 00000002, type 01000000 - size c000
  Name: dwmapi.dll
MZ at 74120000, prot 00000002, type 01000000 - size 14000
  Name: MSACM32.dll
MZ at 74140000, prot 00000002, type 01000000 - size 66000
  Name: audioeng.dll
MZ at 74240000, prot 00000002, type 01000000 - size 7000
  Name: MIDIMAP.dll
MZ at 74260000, prot 00000002, type 01000000 - size 9000
  Name: MSACM32.DRV
MZ at 742a0000, prot 00000002, type 01000000 - size 21000
  Name: AudioSes.DLL
MZ at 742d0000, prot 00000002, type 01000000 - size 2f000
  Name: WINMMDRV.dll
MZ at 74300000, prot 00000002, type 01000000 - size bb000
  Name: PROPSYS.dll
MZ at 743e0000, prot 00000002, type 01000000 - size 8000
  Name: WINRNR.dll
MZ at 743f0000, prot 00000002, type 01000000 - size c000
  Name: wshbth.dll
MZ at 74400000, prot 00000002, type 01000000 - size 3d000
  Name: OLEACC.dll
MZ at 744e0000, prot 00000002, type 01000000 - size 14000
  Name: ATL.DLL
MZ at 74500000, prot 00000002, type 01000000 - size 32000
  Name: WINMM.dll
MZ at 74570000, prot 00000002, type 01000000 - size 6000
```

```
  Name: DCIMAN32.dll
MZ at 74580000, prot 00000002, type 01000000 - size 1ab000
  Name: gdiplus.dll
MZ at 748a0000, prot 00000002, type 01000000 - size f000
  Name: NAPINSP.dll
MZ at 74bd0000, prot 00000002, type 01000000 - size 19e000
  Name: COMCTL32.dll
MZ at 74d70000, prot 00000002, type 01000000 - size f000
  Name: nlaapi.dll
MZ at 74db0000, prot 00000002, type 01000000 - size 28000
  Name: MMDevAPI.DLL
MZ at 74e40000, prot 00000002, type 01000000 - size 15000
  Name: Cabinet.dll
MZ at 74e80000, prot 00000002, type 01000000 - size 4000
  Name: ksuser.dll
MZ at 74e90000, prot 00000002, type 01000000 - size 7000
  Name: AVRT.dll
MZ at 74ed0000, prot 00000002, type 01000000 - size 3f000
  Name: UxTheme.dll
MZ at 74f60000, prot 00000002, type 01000000 - size 2d000
  Name: WINTRUST.dll
MZ at 75140000, prot 00000002, type 01000000 - size 5000
  Name: WSHTCPIP.dll
MZ at 75150000, prot 00000002, type 01000000 - size 5000
  Name: MSIMG32.dll
MZ at 75160000, prot 00000002, type 01000000 - size 1a000
  Name: POWRPROF.dll
MZ at 75180000, prot 00000002, type 01000000 - size 21000
  Name: NTMARTA.dll
MZ at 751e0000, prot 00000002, type 01000000 - size 15000
  Name: GPAPI.dll
MZ at 75220000, prot 00000002, type 01000000 - size 3b000
  Name: RSAENH.dll
MZ at 75260000, prot 00000002, type 01000000 - size 46000
  Name: SCHANNEL.dll
MZ at 75470000, prot 00000002, type 01000000 - size 3b000
  Name: MSWSOCK.dll
MZ at 754e0000, prot 00000002, type 01000000 - size 5000
  Name: WSHIP6.dll
MZ at 75570000, prot 00000002, type 01000000 - size 45000
  Name: bcrypt.dll
MZ at 755c0000, prot 00000002, type 01000000 - size 35000
  Name: ncrypt.dll
MZ at 75610000, prot 00000002, type 01000000 - size 8000
  Name: VERSION.dll
MZ at 75630000, prot 00000002, type 01000000 - size 7000
  Name: CREDSSP.dll
MZ at 75670000, prot 00000002, type 01000000 - size 22000
  Name: dhcpcsvc6.DLL
MZ at 756a0000, prot 00000002, type 01000000 - size 7000
  Name: WINNSI.DLL
MZ at 756b0000, prot 00000002, type 01000000 - size 35000
  Name: dhcpcsvc.DLL
MZ at 756f0000, prot 00000002, type 01000000 - size 19000
  Name: IPHLPAPI.DLL
MZ at 75750000, prot 00000002, type 01000000 - size 3a000
  Name: slc.dll
MZ at 75790000, prot 00000002, type 01000000 - size f2000
  Name: CRYPT32.dll
MZ at 758f0000, prot 00000002, type 01000000 - size 12000
```

```
  Name: MSASN1.dll
MZ at 75930000, prot 00000002, type 01000000 - size 11000
  Name: SAMLIB.dll
MZ at 759a0000, prot 00000002, type 01000000 - size 76000
  Name: NETAPI32.dll
MZ at 75a20000, prot 00000002, type 01000000 - size 2c000
  Name: DNSAPI.dll
MZ at 75c30000, prot 00000002, type 01000000 - size 5f000
  Name: sxs.dll
MZ at 75c90000, prot 00000002, type 01000000 - size 2c000
  Name: apphelp.dll
MZ at 75cf0000, prot 00000002, type 01000000 - size 14000
  Name: Secur32.dll
MZ at 75d10000, prot 00000002, type 01000000 - size 1e000
  Name: USERENV.dll
MZ at 75e50000, prot 00000002, type 01000000 - size 7000
  Name: PSAPI.DLL
MZ at 75e60000, prot 00000002, type 01000000 - size 6000
  Name: NSI.dll
MZ at 75e70000, prot 00000002, type 01000000 - size c3000
  Name: RPCRT4.dll
MZ at 75f40000, prot 00000002, type 01000000 - size 18a000
  Name: SETUPAPI.dll
MZ at 760d0000, prot 00000002, type 01000000 - size 9000
  Name: LPK.dll
MZ at 760e0000, prot 00000002, type 01000000 - size 145000
  Name: ole32.dll
MZ at 76230000, prot 00000002, type 01000000 - size dc000
  Name: KERNEL32.dll
MZ at 76310000, prot 00000002, type 01000000 - size 1e8000
  Name: iertutil.dll
MZ at 76500000, prot 00000002, type 01000000 - size 8d000
  Name: OLEAUT32.dll
MZ at 76590000, prot 00000002, type 01000000 - size 133000
  Name: urlmon.dll
MZ at 766d0000, prot 00000002, type 01000000 - size b10000
  Name: SHELL32.dll
MZ at 771e0000, prot 00000002, type 01000000 - size 84000
  Name: CLBCatQ.DLL
MZ at 77270000, prot 00000002, type 01000000 - size aa000
  Name: msvcrt.dll
MZ at 77320000, prot 00000002, type 01000000 - size 59000
  Name: SHLWAPI.dll
MZ at 77380000, prot 00000002, type 01000000 - size 2d000
  Name: WS2_32.dll
MZ at 773b0000, prot 00000002, type 01000000 - size 4b000
  Name: GDI32.dll
MZ at 77400000, prot 00000002, type 01000000 - size 9d000
  Name: USER32.dll
MZ at 774a0000, prot 00000002, type 01000000 - size 73000
  Name: COMDLG32.dll
MZ at 77520000, prot 00000002, type 01000000 - size 49000
  Name: WLDAP32.dll
MZ at 77570000, prot 00000002, type 01000000 - size e6000
  Name: WININET.dll
MZ at 77660000, prot 00000002, type 01000000 - size 7d000
  Name: USP10.dll
MZ at 776e0000, prot 00000002, type 01000000 - size c6000
  Name: ADVAPI32.dll
MZ at 777b0000, prot 00000002, type 01000000 - size 127000
```

```
  Name: ntdll.dll
MZ at 778e0000, prot 00000002, type 01000000 - size 3000
  Name: Normaliz.dll
MZ at 778f0000, prot 00000002, type 01000000 - size 1e000
  Name: IMM32.dll
MZ at 77910000, prot 00000002, type 01000000 - size 29000
  Name: imagehlp.dll
MZ at 77940000, prot 00000002, type 01000000 - size c8000
  Name: MSCTF.dll
MZ at 7c340000, prot 00000002, type 01000000 - size 56000
  Name: MSVCR71.dll
MZ at 7c3a0000, prot 00000002, type 01000000 - size 7b000
  Name: MSVCP71.dll
```

We see *screens_dll.dll* module with READWRITE protection attribute different from all other found modules:

```
0:004> !address 10000000

Usage:                  <unknown>
Base Address:           10000000
End Address:            10001000
Region Size:            00001000 (   4.000 kB)
State:                  00001000          MEM_COMMIT
Protect:                00000004          PAGE_READWRITE
Type:                   00020000          MEM_PRIVATE
Allocation Base:        10000000
Allocation Protect:     00000004          PAGE_READWRITE


Content source: 1 (target), length: 1000
```

11. We now check module headers for this DLL:

```
0:004> !dh 10000000

File Type: DLL
FILE HEADER VALUES
     14C machine (i386)
       4 number of sections
4C8FEE9E time date stamp Tue Sep 14 22:52:30 2010

       0 file pointer to symbol table
       0 number of symbols
      E0 size of optional header
    2102 characteristics
            Executable
            32 bit word machine
            DLL

OPTIONAL HEADER VALUES
     10B magic #
    9.00 linker version
     400 size of code
     800 size of initialized data
       0 size of uninitialized data
    12F3 address of entry point
    1000 base of code
         ----- new -----
10000000 image base
```

```
    1000 section alignment
     200 file alignment
       2 subsystem (Windows GUI)
    5.00 operating system version
    0.00 image version
    5.00 subsystem version
    5000 size of image
     400 size of headers
       0 checksum
00100000 size of stack reserve
00001000 size of stack commit
00100000 size of heap reserve
00001000 size of heap commit
     140  DLL characteristics
            Dynamic base
            NX compatible
    2330 [      50] address [size] of Export Directory
    20E0 [      78] address [size] of Import Directory
       0 [       0] address [size] of Resource Directory
       0 [       0] address [size] of Exception Directory
       0 [       0] address [size] of Security Directory
    4000 [      34] address [size] of Base Relocation Directory
    2060 [      1C] address [size] of Debug Directory
       0 [       0] address [size] of Description Directory
       0 [       0] address [size] of Special Directory
       0 [       0] address [size] of Thread Storage Directory
       0 [       0] address [size] of Load Configuration Directory
       0 [       0] address [size] of Bound Import Directory
    2000 [      58] address [size] of Import Address Table Directory
       0 [       0] address [size] of Delay Import Directory
       0 [       0] address [size] of COR20 Header Directory
       0 [       0] address [size] of Reserved Directory


SECTION HEADER #1
   .text name
10001000 virtual size
    1000 virtual address
     400 size of raw data
     400 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
60000020 flags
         Code
         (no align specified)
         Execute Read

SECTION HEADER #2
  .rdata name
10002000 virtual size
    2000 virtual address
     400 size of raw data
     800 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
40000040 flags
```

```
         Initialized Data
         (no align specified)
         Read Only


Debug Directories(1)
       Type       Size     Address  Pointer
       cv          46        2094      894    Format: RSDS, guid, 1,
C:\MyWork\screens_dll\Release\screens_dll.pdb

SECTION HEADER #3
   .data name
10003000 virtual size
    3000 virtual address
       0 size of raw data
       0 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
C0000040 flags
         Initialized Data
         (no align specified)
         Read Write

SECTION HEADER #4
  .reloc name
10004000 virtual size
    4000 virtual address
     200 size of raw data
     C00 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
42000040 flags
         Initialized Data
         Discardable
         (no align specified)
         Read Only
```

It looks like a normal DLL but its import address table reveals its purpose - screen capture:

```
0:004> dps 10000000+2000 L58/4
10002000  773b6101 gdi32!CreateCompatibleDC
10002004  773b93d6 gdi32!StretchBlt
10002008  773b7461 gdi32!CreateDIBSection
1000200c  773b62a0 gdi32!SelectObject
10002010  00000000
10002014  7627a411 kernel32!lstrcmpW
10002018  762740aa kernel32!VirtualFree
1000201c  7627ad55 kernel32!VirtualAlloc
10002020  00000000
10002024  77419ced user32!ReleaseDC
10002028  77413ba7 user32!NtUserGetWindowDC
1000202c  77420e21 user32!GetWindowRect
10002030  00000000
10002034  745975e9 GdiPlus!GdiplusStartup
10002038  745876dd GdiPlus!GdipSaveImageToStream
1000203c  745bdd38 GdiPlus!GdipGetImageEncodersSize
10002040  745871cf GdiPlus!GdipDisposeImage
```

```
10002044  74598591 GdiPlus!GdipCreateBitmapFromHBITMAP
10002048  745bdbae GdiPlus!GdipGetImageEncoders
1000204c  00000000
10002050  7613d51b ole32!CreateStreamOnHGlobal [d:\longhorn\com\ole32\ole232\base\memstm.cpp @ 1518]
10002054  00000000
```

12. And finally, heap analysis of a corrupt entry reveals the captured password:

```
0:004> !heap -s -v
SEGMENT HEAP ERROR: failed to initialize the extention
**************************************************************
*                                                            *
*                   HEAP ERROR DETECTED                      *
*                                                            *
**************************************************************

Details:

Heap address:  00290000
Error address: 04f1ffe0
Error type:    HEAP_FAILURE_ENTRY_CORRUPTION
Details:       The heap manager detected a corrupt heap entry.
Follow-up:     Enable pageheap.

Stack trace:
                7782b1a5: ntdll!RtlpCoalesceFreeBlocks+0x000004b9
                7781730a: ntdll!RtlpFreeHeap+0x000001e2
                77817545: ntdll!RtlFreeHeap+0x0000014e
                76277e4b: kernel32!GlobalFree+0x00000047
                760f7277: ole32!ReleaseStgMedium+0x00000124
                76594a1f: urlmon!ReleaseBindInfo+0x0000004c
                765f7feb: urlmon!CINet::ReleaseCNetObjects+0x0000003d
                765b9a87: urlmon!CINetHttp::OnWininetRequestHandleClosing+0x00000060
                765b93f0: urlmon!CINet::CINetCallback+0x000002de
                77582078: wininet!InternetIndicateStatus+0x000000fc
                77588f5d: wininet!HANDLE_OBJECT::~HANDLE_OBJECT+0x000000c9
                7758937a: wininet!INTERNET_CONNECT_HANDLE_OBJECT::~INTERNET_CONNECT_HANDLE_OBJECT+0x00000209
                7758916b: wininet!HTTP_REQUEST_HANDLE_OBJECT::`scalar deleting destructor'+0x0000000d
                77588d5e: wininet!HANDLE_OBJECT::Dereference+0x00000022
                77589419: wininet!_InternetCloseHandle+0x0000009d
                77589114: wininet!InternetCloseHandle+0x0000011e

[...]

0:004> dc 04f1ffe0-20
04f1ffc0  6161613d 61616161 26616161 50747874  =aaaaaaaaaa&txtP
04f1ffd0  77737361 3d64726f 61616161 61616161  assword=aaaaaaaa
04f1ffe0  74933b00 0310f0ba 00000000 00000000  .;.t............
04f1fff0  04e20038 04e20038 04f20000 00000000  8...8...........
04f20000  ???????? ???????? ???????? ????????  ????????????????
04f20010  ???????? ???????? ???????? ????????  ????????????????
04f20020  ???????? ???????? ???????? ????????  ????????????????
04f20030  ???????? ???????? ???????? ????????  ????????????????
```

13. We should also check for any patched module code in all modules to which we have matching file binary access (if you use a docker environment, please specify this command **.exepath** *C:\mss* before):

```
0:004> !for_each_module "!chkimg -v -d @#ModuleName"

[...]

Scanning section:    .text
Size: 1307933
Range to scan: 74bd1000-74d1051d
    74ca8814-74ca8818  5 bytes - comctl32!PropertySheetW
        [ 8b ff 55 8b ec:e9 e8 d8 d9 fb ]
    74ca882c-74ca8830  5 bytes - comctl32!PropertySheetA (+0x18)
```

```
       [ 8b ff 55 8b ec:e9 70 d9 d9 fb ]
Total bytes compared: 1307933(100%)
Number of errors: 10

[...]

Scanning section:    .text
Size: 1204234
Range to scan: 760e1000-7620700a
    76101e12-76101e16  5 bytes - ole32!OleLoadFromStream
       [ 8b ff 55 8b ec:e9 b9 30 94 fa ]
    76139ea6-76139eaa  5 bytes - ole32!CoCreateInstance (+0x38094)
       [ 8b ff 55 8b ec:e9 d5 3c 81 fa ]
Total bytes compared: 1204234(100%)
Number of errors: 10

[...]

Scanning section:    .text
Size: 528293
Range to scan: 76501000-76581fa5
    76503df0-76503df4  5 bytes - oleaut32!VariantClear
       [ 8b ff 55 8b ec:e9 1f 1d 54 fa ]
    76503e40-76503e44  5 bytes - oleaut32!SysFreeString (+0x50)
       [ 8b ff 55 8b ec:e9 f3 10 54 fa ]
    7650462b-7650462f  5 bytes - oleaut32!SysAllocStringByteLen (+0x7eb)
       [ 8b ff 55 8b ec:e9 4a 14 54 fa ]
    765074bc-765074c0  5 bytes - oleaut32!VariantChangeType (+0x2e91)
       [ 8b ff 55 8b ec:e9 04 e6 53 fa ]
    765670ae-765670b2  5 bytes - oleaut32!OleCreatePropertyFrameIndirect (+0x5fbf2)
       [ 8b ff 55 8b ec:e9 96 e6 4d fa ]
Total bytes compared: 528293(100%)
Number of errors: 25

[...]

Scanning section:    .text
Size: 3612636
Range to scan: 766d1000-76a42fdc
    767589a8-767589ab  4 bytes - shell32!CRegFolder::`vftable'
       [ 88 20 76 76:4d 30 c1 6d ]
    767589b0-767589b7  8 bytes - shell32!CRegFolder::`vftable'+8 (+0x08)
       [ 2f 92 75 76 df e4 75 76:57 2f c1 6d 9c 5b c0 6d ]
Total bytes compared: 3612636(100%)
Number of errors: 12

[...]

Scanning section:    .text
Size: 422527
Range to scan: 77401000-7746827f
    774072a2-774072a6  5 bytes - user32!CreateDialogParamW
       [ 8b ff 55 8b ec:e9 09 6c 54 f9 ]
    7740863c-77408640  5 bytes - user32!GetAsyncKeyState (+0x139a)
       [ 8b ff 55 8b ec:e9 f6 08 46 f9 ]
    774087ad-774087b1  5 bytes - user32!SetWindowsHookExW (+0x171)
       [ 8b ff 55 8b ec:e9 23 13 54 f9 ]
    77408e3b-77408e3f  5 bytes - user32!CallNextHookEx (+0x68e)
       [ 8b ff 55 8b ec:e9 f5 42 53 f9 ]
    774098db-774098df  5 bytes - user32!NtUserUnhookWindowsHookEx (+0xaa0)
```

```
      [ b8 52 12 00 00:e9 86 ad 4a f9 ]
    7740cd8b-7740cd8f  5 bytes - user32!EnableWindow (+0x34b0)
      [ 8b ff 55 8b ec:e9 ad 0f 54 f9 ]
    77411305-77411309  5 bytes - user32!CreateWindowExW (+0x457a)
      [ 8b ff 55 8b ec:e9 1a c8 53 f9 ]
    77418cb1-77418cb5  5 bytes - user32!GetKeyState (+0x79ac)
      [ 8b ff 55 8b ec:e9 35 46 53 f9 ]
    77420745-77420749  5 bytes - user32!IsDialogMessageW (+0x7a94)
      [ 8b ff 55 8b ec:e9 c9 52 45 f9 ]
    774217aa-774217ae  5 bytes - user32!CreateDialogParamA (+0x1065)
      [ 8b ff 55 8b ec:e9 27 40 62 f9 ]
    77421847-7742184b  5 bytes - user32!IsDialogMessageA (+0x9d)
      [ 8b ff 55 8b ec:e9 26 38 62 f9 ]
    774226f1-774226f5  5 bytes - user32!CreateDialogIndirectParamA (+0xeaa)
      [ 8b ff 55 8b ec:e9 17 31 62 f9 ]
    77429a62-77429a66  5 bytes - user32!CreateDialogIndirectParamW (+0x7371)
      [ 8b ff 55 8b ec:e9 dd bd 61 f9 ]
    77430987-7743098b  5 bytes - user32!NtUserSetKeyboardState (+0x6f25)
      [ b8 20 12 00 00:e9 55 4a 61 f9 ]
    774310b0-774310b4  5 bytes - user32!DialogBoxParamW (+0x729)
      [ 8b ff 55 8b ec:e9 4c 44 44 f9 ]
    77432ef5-77432ef9  5 bytes - user32!DialogBoxIndirectParamW (+0x1e45)
      [ 8b ff 55 8b ec:e9 55 1c 61 f9 ]
    77432f75-77432f79  5 bytes - user32!NtUserSendInput (+0x80)
      [ b8 0d 12 00 00:e9 25 30 61 f9 ]
    7743326e-77433272  5 bytes - user32!EndDialog (+0x2f9)
      [ 8b ff 55 8b ec:e9 47 4c 44 f9 ]
    77446fb2-77446fb6  5 bytes - user32!SetCursorPos (+0x13d44)
      [ 8b ff 55 8b ec:e9 3c f0 5f f9 ]
    77448152-77448156  5 bytes - user32!DialogBoxParamA (+0x11a0)
      [ 8b ff 55 8b ec:e9 95 c9 5f f9 ]
    7744847d-77448481  5 bytes - user32!DialogBoxIndirectParamA (+0x32b)
      [ 8b ff 55 8b ec:e9 30 c7 5f f9 ]
    7745d4d9-7745d4dd  5 bytes - user32!MessageBoxIndirectA (+0x1505c)
      [ 8b ff 55 8b ec:e9 a3 75 5e f9 ]
    7745d5d3-7745d5d7  5 bytes - user32!MessageBoxIndirectW (+0xfa)
      [ 8b ff 55 8b ec:e9 3e 74 5e f9 ]
    7745d639-7745d63d  5 bytes - user32!MessageBoxExA (+0x66)
      [ 8b ff 55 8b ec:e9 76 73 5e f9 ]
    7745d65d-7745d661  5 bytes - user32!MessageBoxExW (+0x24)
      [ 8b ff 55 8b ec:e9 f0 72 5e f9 ]
    7745d972-7745d976  5 bytes - user32!keybd_event (+0x315)
      [ 8b ff 55 8b ec:e9 ac 89 5e f9 ]
Total bytes compared: 422527(100%)
Number of errors: 130

[...]

Scanning section:    .text
Size: 320529
Range to scan: 774a1000-774ef411
    774a30cf-774a30d3  5 bytes - comdlg32!PrintDlgW
      [ 8b ff 55 8b ec:e9 41 28 5a f9 ]
    774ced29-774ced2d  5 bytes - comdlg32!PageSetupDlgW (+0x2bc5a)
      [ 8b ff 55 8b ec:e9 4d 6b 57 f9 ]
Total bytes compared: 320529(100%)
Number of errors: 10
10 errors : comdlg32 (774a30cf-774ced2d)

[...]
```

```
Scanning section:    .text
Size: 794010
Range to scan: 777b1000-77872d9a
    77814dba-77814dbd  4 bytes - ntdll!ZwQueryDirectoryFile+6
      [ 00 03 fe 7f:e8 af 05 00 ]
    778151ba-778151bd  4 bytes - ntdll!ZwResumeThread+6 (+0x400)
      [ 00 03 fe 7f:d8 af 05 00 ]
Total bytes compared: 794010(100%)
Number of errors: 8

[...]
```

When we look at the reported patched address, we find out that most of them belong to IE:

```
0:004> u 774a30cf
comdlg32!PrintDlgW:
774a30cf e941285af9      jmp     ieframe!Detour_PrintDlgW (70a45915)
774a30d4 81eca0040000    sub     esp,4A0h
774a30da a1ac034f77      mov     eax,dword ptr [comdlg32!__security_cookie (774f03ac)]
774a30df 33c5            xor     eax,ebp
774a30e1 8945fc          mov     dword ptr [ebp-4],eax
774a30e4 56              push    esi
774a30e5 8b7508          mov     esi,dword ptr [ebp+8]
774a30e8 689c040000      push    49Ch
```

However, the last two addresses are suspicious as they do not belong to IE and show “garbage”:

```
0:004> u 77814dba
ntdll!ZwQueryDirectoryFile+0x6:
77814dba e8af0500ff      call    shell32!MetadataLayout::UpdateDesiredSize+0x218 (7681536e)
77814dbf 12c2            adc     al,dl
77814dc1 2c00            sub     al,0
77814dc3 90              nop
ntdll!NtQueryDirectoryObject:
77814dc4 b8db000000      mov     eax,0DBh
77814dc9 ba0003fe7f      mov     edx,offset SharedUserData!SystemCallStub (7ffe0300)
77814dce ff12            call    dword ptr [edx]
77814dd0 c21c00          ret     1Ch

0:004> u 7681536e
shell32!MetadataLayout::UpdateDesiredSize+0x218:
7681536e 46              inc     esi
7681536f 18894df80f82    sbb     byte ptr [ecx-7DF007B3h],cl
76815375 51              push    ecx
76815376 ff              ???
76815377 ff              ???
76815378 ff8b46288b55    dec     dword ptr [ebx+558B2846h]
7681537e 108d04988b08    adc     byte ptr [ebp+88B9804h],cl
76815384 014df0          add     dword ptr [ebp-10h],ecx

0:004> ub 77814dba
                 ^ Unable to find valid previous instruction for 'ub 77814dba'
```

Here we needed to check the beginning of the function because the patching may be done for the part of an instruction such as changing an address or an offset:

```
0:004> u ntdll!ZwQueryDirectoryFile
ntdll!ZwQueryDirectoryFile:
77814db4 b8da000000      mov     eax,0DAh
77814db9 bae8af0500      mov     edx,5AFE8h
77814dbe ff12            call    dword ptr [edx]
77814dc0 c22c00          ret     2Ch
77814dc3 90              nop
ntdll!NtQueryDirectoryObject:
77814dc4 b8db000000      mov     eax,0DBh
77814dc9 ba0003fe7f      mov     edx,offset SharedUserData!SystemCallStub (7ffe0300)
77814dce ff12            call    dword ptr [edx]
```

Note that a pointer to an indirect call has changed: in the normal case, we see this:

```
0:004> dps 7ffe0300 L1
7ffe0300  77815e70 ntdll!KiFastSystemCall
```

In the abnormal case, we have execution diversion to already discovered malware module:

```
0:004> dps 5AFE8h L1
0005afe8  0004efe0

0:004> u 0004efe0
0004efe0 58              pop     eax
0004efe1 8d0510ec0400    lea     eax,ds:[4EC10h]
0004efe7 ffe0            jmp     eax
0004efe9 c3              ret
0004efea cc              int     3
0004efeb cc              int     3
0004efec cc              int     3
0004efed cc              int     3

0:004> u 4EC10h
0004ec10 55              push    ebp
0004ec11 8bec            mov     ebp,esp
0004ec13 83ec38          sub     esp,38h
0004ec16 0fb64530        movzx   eax,byte ptr [ebp+30h]
0004ec1a 50              push    eax
0004ec1b 8b4d2c          mov     ecx,dword ptr [ebp+2Ch]
0004ec1e 51              push    ecx
0004ec1f 0fb65528        movzx   edx,byte ptr [ebp+28h]

0:004> !address 4EC10h


Mapping file section regions...
Mapping module regions...
Mapping PEB regions...
Mapping TEB and stack regions...
Mapping heap regions...
*** Failure in mapping Heap (80004005: ExtRemoteTyped::Field: unable to retrieve field
'BaseAddress' at ffffffff99654a5f)
Mapping page heap regions...
Mapping other regions...
Mapping stack trace database regions...
Mapping activation context regions...

Usage:                  <unknown>
```

```
Base Address:           00040000
End Address:            0005d000
Region Size:            0001d000 ( 116.000 kB)
State:                  00001000          MEM_COMMIT
Protect:                00000040          PAGE_EXECUTE_READWRITE
Type:                   00020000          MEM_PRIVATE
Allocation Base:        00040000
Allocation Protect:     00000040          PAGE_EXECUTE_READWRITE


Content source: 1 (target), length: e3f0
```

Note that here we have execution redirection based on system call dispatch. This is a different pathway than patching **Import Address Table** functions. Here ntdll!Zw* functions are meant to transition to kernel space to execute corresponding system services there. This transition is commonly done through the pseudo module SharedUserData:

```
0:004> !address SharedUserData

Usage:                  Other
Base Address:           7ffe0000
End Address:            7ffe1000
Region Size:            00001000 (   4.000 kB)
State:                  00001000          MEM_COMMIT
Protect:                00000002          PAGE_READONLY
Type:                   00020000          MEM_PRIVATE
Allocation Base:        7ffe0000
Allocation Protect:     00000002          PAGE_READONLY
Additional info:        User Shared Data


Content source: 1 (target), length: 1000

0:004> dps SharedUserData!SystemCallStub L1
7ffe0300  77815e70 ntdll!KiFastSystemCall

0:004> uf ntdll!KiFastSystemCall
ntdll!KiFastSystemCall:
77815e70 8bd4            mov     edx,esp
77815e72 0f34            sysenter
77815e74 c3              ret
```

14. Another check is for exception handlers. We can check the current problem thread or for all threads via **~*e** command. Note that an exception can happen on each thread, each having different handlers.

```
0:004> !exchain
02c9cb90: ntdll!_except_handler4+0 (777b99fa)
  CRT scope  0, func:   ntdll!RtlReportExceptionEx+187 (77843ca3)
02c9cbd0: ntdll!_except_handler4+0 (777b99fa)
  CRT scope  0, filter: ntdll!RtlReportException+53 (77843d67)
                func:   ntdll!RtlReportException+57 (77843d70)
02c9cc54: ntdll!ExecuteHandler2+3a (77815f8d)
02c9d074: ntdll!_except_handler4+0 (777b99fa)
  CRT scope  0, filter: ntdll!RtlReportCriticalFailure+5d (7785faff)
                func:   ntdll!RtlReportCriticalFailure+6c (7785fb13)
02c9d0b8: ntdll!_except_handler4+0 (777b99fa)
  CRT scope  0, filter: ntdll!RtlpLogHeapFailure+83 (778607cf)
                func:   ntdll!RtlpLogHeapFailure+90 (778607e1)
02c9d1f8: ntdll!_except_handler4+0 (777b99fa)
```

```
  CRT scope  0, func:   ntdll!RtlpFreeHeap+b0c (7782b9f7)
02c9d25c: kernel32!_except_handler4+0 (7626fd89)
  CRT scope  0, filter: kernel32!GlobalFree+11c (7628e1e7)
                func:   kernel32!GlobalFree+133 (7628e203)
02c9d6f0: user32!_except_handler4+0 (7746522d)
  CRT scope  0, func:   user32!UserCallWinProcCheckWow+150 (77436e2c)
02c9d754: user32!_except_handler4+0 (7746522d)
  CRT scope  0, filter: user32!DispatchMessageWorker+144 (77437cbc)
                func:   user32!DispatchMessageWorker+157 (77437cd4)
02c9f980: ntdll!_except_handler4+0 (777b99fa)
  CRT scope  0, filter: ntdll!__RtlUserThreadStart+3b (77827f8d)
                func:   ntdll!__RtlUserThreadStart+70 (77827fc7)
Invalid exception stack at ffffffff

0:004> ~*e !exchain
001df568: kernel32!_except_handler4+0 (7626fd89)
  CRT scope  1, func:   kernel32!WaitForMultipleObjectsEx+18a (7627a628)
  CRT scope  0, func:   kernel32!WaitForMultipleObjectsEx+186 (7627a630)
001df85c: iexplore!_except_handler4+0 (00ff6944)
  CRT scope  1, filter: iexplore!_initterm_e+1da (00ff3153)
                func:   iexplore!_initterm_e+1ee (00ff316c)
001df8a8: ntdll!_except_handler4+0 (777b99fa)
  CRT scope  0, filter: ntdll!__RtlUserThreadStart+3b (77827f8d)
                func:   ntdll!__RtlUserThreadStart+70 (77827fc7)
Invalid exception stack at ffffffff
0258f860: ntdll!_except_handler4+0 (777b99fa)
  CRT scope  2, func:   ntdll!TppWaiterpThread+63c (7783a9bb)
  CRT scope  1, func:   ntdll!TppWaiterpThread+6e9 (777c098e)
  CRT scope  0, filter: ntdll!TppWaiterpThread+6f2 (7783aa39)
                func:   ntdll!TppWaiterpThread+703 (7783aa4f)
0258f8ac: ntdll!_except_handler4+0 (777b99fa)
  CRT scope  0, filter: ntdll!__RtlUserThreadStart+3b (77827f8d)
                func:   ntdll!__RtlUserThreadStart+70 (77827fc7)
Invalid exception stack at ffffffff
02a2edcc: kernel32!_except_handler4+0 (7626fd89)
  CRT scope  1, func:   kernel32!WaitForMultipleObjectsEx+18a (7627a628)
  CRT scope  0, func:   kernel32!WaitForMultipleObjectsEx+186 (7627a630)
02a2fe68: ntdll!_except_handler4+0 (777b99fa)
  CRT scope  0, filter: ntdll!__RtlUserThreadStart+3b (77827f8d)
                func:   ntdll!__RtlUserThreadStart+70 (77827fc7)
Invalid exception stack at ffffffff
028efa38: kernel32!_except_handler4+0 (7626fd89)
  CRT scope  1, func:   kernel32!WaitForMultipleObjectsEx+18a (7627a628)
  CRT scope  0, func:   kernel32!WaitForMultipleObjectsEx+186 (7627a630)
028efb48: ntdll!_except_handler4+0 (777b99fa)
  CRT scope  0, filter: ntdll!__RtlUserThreadStart+3b (77827f8d)
                func:   ntdll!__RtlUserThreadStart+70 (77827fc7)
Invalid exception stack at ffffffff
02c9cb90: ntdll!_except_handler4+0 (777b99fa)
  CRT scope  0, func:   ntdll!RtlReportExceptionEx+187 (77843ca3)
02c9cbd0: ntdll!_except_handler4+0 (777b99fa)
  CRT scope  0, filter: ntdll!RtlReportException+53 (77843d67)
                func:   ntdll!RtlReportException+57 (77843d70)
02c9cc54: ntdll!ExecuteHandler2+3a (77815f8d)
02c9d074: ntdll!_except_handler4+0 (777b99fa)
  CRT scope  0, filter: ntdll!RtlReportCriticalFailure+5d (7785faff)
                func:   ntdll!RtlReportCriticalFailure+6c (7785fb13)
02c9d0b8: ntdll!_except_handler4+0 (777b99fa)
  CRT scope  0, filter: ntdll!RtlpLogHeapFailure+83 (778607cf)
                func:   ntdll!RtlpLogHeapFailure+90 (778607e1)
```

```
02c9d1f8: ntdll!_except_handler4+0 (777b99fa)
  CRT scope  0, func:   ntdll!RtlpFreeHeap+b0c (7782b9f7)
02c9d25c: kernel32!_except_handler4+0 (7626fd89)
  CRT scope  0, filter: kernel32!GlobalFree+11c (7628e1e7)
                func:   kernel32!GlobalFree+133 (7628e203)
02c9d6f0: user32!_except_handler4+0 (7746522d)
  CRT scope  0, func:   user32!UserCallWinProcCheckWow+150 (77436e2c)
02c9d754: user32!_except_handler4+0 (7746522d)
  CRT scope  0, filter: user32!DispatchMessageWorker+144 (77437cbc)
                func:   user32!DispatchMessageWorker+157 (77437cd4)
02c9f980: ntdll!_except_handler4+0 (777b99fa)
  CRT scope  0, filter: ntdll!__RtlUserThreadStart+3b (77827f8d)
                func:   ntdll!__RtlUserThreadStart+70 (77827fc7)
Invalid exception stack at ffffffff
02e8fad8: kernel32!_except_handler4+0 (7626fd89)
  CRT scope  1, func:   kernel32!WaitForMultipleObjectsEx+18a (7627a628)
  CRT scope  0, func:   kernel32!WaitForMultipleObjectsEx+186 (7627a630)
02e8fc40: msidcrl40!CreatePassportAuthUIContext+5e13b (275f8bcb)
02e8fc78: ntdll!_except_handler4+0 (777b99fa)
  CRT scope  0, func:   ntdll!RtlpTpTimerCallback+8e (7783b037)
02e8fdcc: ntdll!_except_handler4+0 (777b99fa)
  CRT scope  8, filter: ntdll!TppWorkerThread+515 (77839f8d)
                func:   ntdll!TppWorkerThread+531 (77839fae)
  CRT scope  2, func:   ntdll!TppWorkerThread+6c2 (777e6fdb)
  CRT scope  1, func:   ntdll!TppWorkerThread+78e (777e70cf)
  CRT scope  0, filter: ntdll!TppWorkerThread+79f (7783a09f)
                func:   ntdll!TppWorkerThread+7b4 (7783a0b9)
02e8fe18: ntdll!_except_handler4+0 (777b99fa)
  CRT scope  0, filter: ntdll!__RtlUserThreadStart+3b (77827f8d)
                func:   ntdll!__RtlUserThreadStart+70 (77827fc7)
Invalid exception stack at ffffffff
0409fdcc: kernel32!_except_handler4+0 (7626fd89)
  CRT scope  0, func:   kernel32!SleepEx+91 (76293fa6)
0409fe60: ntdll!_except_handler4+0 (777b99fa)
  CRT scope  0, filter: ntdll!__RtlUserThreadStart+3b (77827f8d)
                func:   ntdll!__RtlUserThreadStart+70 (77827fc7)
Invalid exception stack at ffffffff
03f0fcb8: ntdll!_except_handler4+0 (777b99fa)
  CRT scope  0, filter: ntdll!__RtlUserThreadStart+3b (77827f8d)
                func:   ntdll!__RtlUserThreadStart+70 (77827fc7)
Invalid exception stack at ffffffff
0474f718: mswsock!_except_handler4+0 (7549148b)
  CRT scope  0, filter: mswsock!WSPSelect+52d (7547e749)
                func:   mswsock!WSPSelect+531 (7547e752)
0474f798: ws2_32!_except_handler4+0 (773a24ba)
  CRT scope  0, filter: ws2_32!select+3ba (7738fe6e)
                func:   ws2_32!select+3be (7738fe77)
0474fb44: ntdll!_except_handler4+0 (777b99fa)
  CRT scope  0, filter: ntdll!__RtlUserThreadStart+3b (77827f8d)
                func:   ntdll!__RtlUserThreadStart+70 (77827fc7)
Invalid exception stack at ffffffff
051ff9cc: ntdll!_except_handler4+0 (777b99fa)
  CRT scope  5, filter: ntdll!TppWorkerThread+219 (77839e5c)
                func:   ntdll!TppWorkerThread+230 (77839e78)
  CRT scope  2, func:   ntdll!TppWorkerThread+6c2 (777e6fdb)
  CRT scope  1, func:   ntdll!TppWorkerThread+78e (777e70cf)
  CRT scope  0, filter: ntdll!TppWorkerThread+79f (7783a09f)
                func:   ntdll!TppWorkerThread+7b4 (7783a0b9)
051ffa18: ntdll!_except_handler4+0 (777b99fa)
  CRT scope  0, filter: ntdll!__RtlUserThreadStart+3b (77827f8d)
```

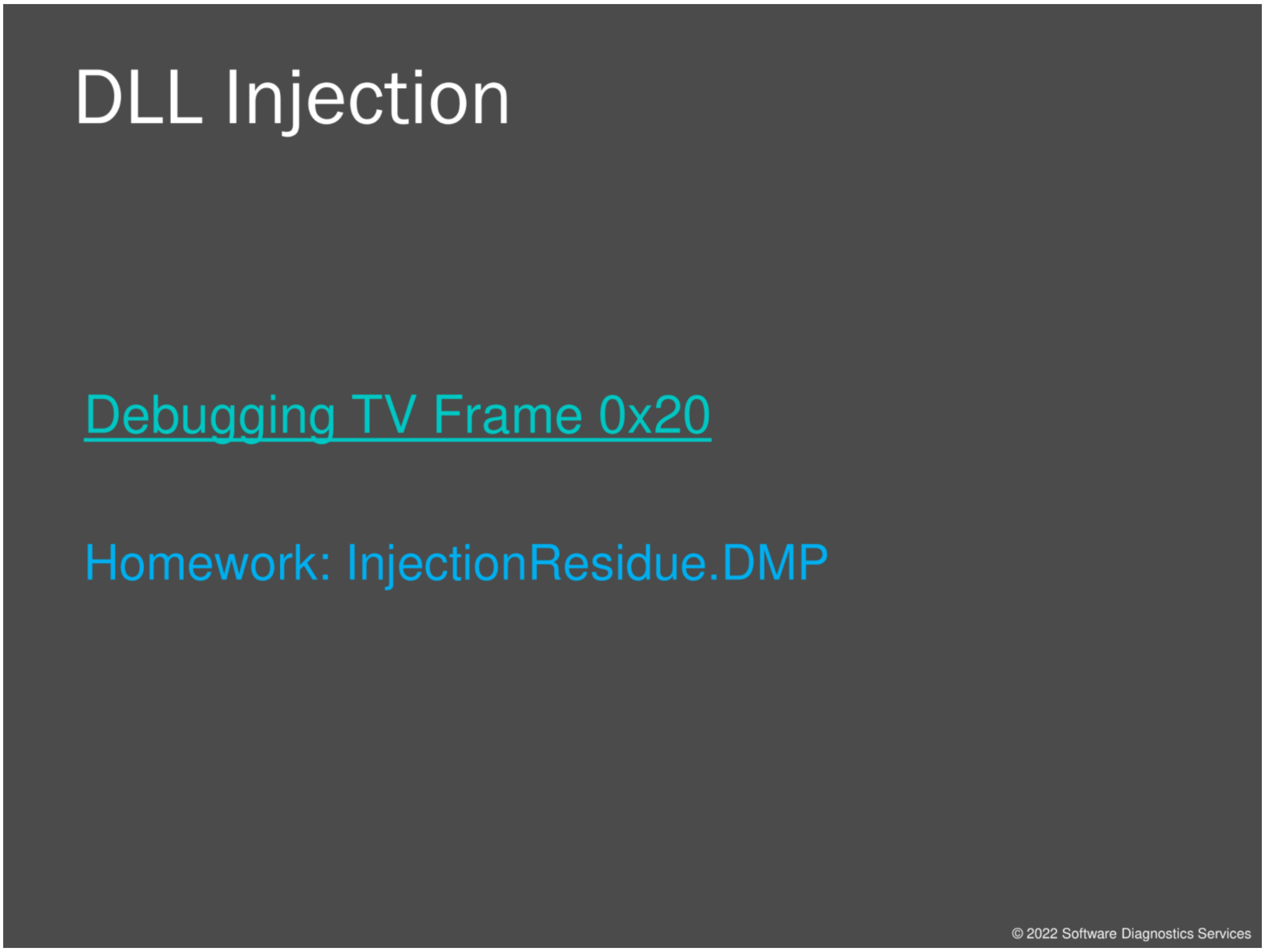
```
                func:   ntdll!__RtlUserThreadStart+70 (77827fc7)
Invalid exception stack at ffffffff
04dbf8d8: ntdll!_except_handler4+0 (777b99fa)
  CRT scope  0, filter: ntdll!__RtlUserThreadStart+3b (77827f8d)
                func:   ntdll!__RtlUserThreadStart+70 (77827fc7)
Invalid exception stack at ffffffff
0568fedc: kernel32!_except_handler4+0 (7626fd89)
  CRT scope  1, func:   kernel32!WaitForSingleObjectEx+fc (762937c7)
  CRT scope  0, func:   kernel32!WaitForSingleObjectEx+110 (762937e2)
0568ff74: ntdll!_except_handler4+0 (777b99fa)
  CRT scope  0, filter: ntdll!__RtlUserThreadStart+3b (77827f8d)
                func:   ntdll!__RtlUserThreadStart+70 (77827fc7)
Invalid exception stack at ffffffff
06e1fd04: kernel32!_except_handler4+0 (7626fd89)
  CRT scope  1, func:   kernel32!WaitForSingleObjectEx+fc (762937c7)
  CRT scope  0, func:   kernel32!WaitForSingleObjectEx+110 (762937e2)
06e1fd9c: ntdll!_except_handler4+0 (777b99fa)
  CRT scope  0, filter: ntdll!__RtlUserThreadStart+3b (77827f8d)
                func:   ntdll!__RtlUserThreadStart+70 (77827fc7)
Invalid exception stack at ffffffff
06f1f76c: kernel32!_except_handler4+0 (7626fd89)
  CRT scope  1, func:   kernel32!WaitForMultipleObjectsEx+18a (7627a628)
  CRT scope  0, func:   kernel32!WaitForMultipleObjectsEx+186 (7627a630)
06f1fad8: msidcrl40!CreatePassportAuthUIContext+5c340 (275f6dd0)
06f1fb0c: msidcrl40!CreatePassportAuthUIContext+2dc00 (275c8690)
06f1fb60: ntdll!_except_handler4+0 (777b99fa)
  CRT scope  0, filter: ntdll!__RtlUserThreadStart+3b (77827f8d)
                func:   ntdll!__RtlUserThreadStart+70 (77827fc7)
Invalid exception stack at ffffffff
0775f68c: kernel32!_except_handler4+0 (7626fd89)
  CRT scope  1, func:   kernel32!WaitForMultipleObjectsEx+18a (7627a628)
  CRT scope  0, func:   kernel32!WaitForMultipleObjectsEx+186 (7627a630)
0775f9f8: msidcrl40!CreatePassportAuthUIContext+5c340 (275f6dd0)
0775fa2c: msidcrl40!CreatePassportAuthUIContext+2dc00 (275c8690)
0775fa80: ntdll!_except_handler4+0 (777b99fa)
  CRT scope  0, filter: ntdll!__RtlUserThreadStart+3b (77827f8d)
                func:   ntdll!__RtlUserThreadStart+70 (77827fc7)
Invalid exception stack at ffffffff
0501fc18: ntdll!_except_handler4+0 (777b99fa)
  CRT scope  5, filter: ntdll!TppWorkerThread+219 (77839e5c)
                func:   ntdll!TppWorkerThread+230 (77839e78)
  CRT scope  2, func:   ntdll!TppWorkerThread+6c2 (777e6fdb)
  CRT scope  1, func:   ntdll!TppWorkerThread+78e (777e70cf)
  CRT scope  0, filter: ntdll!TppWorkerThread+79f (7783a09f)
                func:   ntdll!TppWorkerThread+7b4 (7783a0b9)
0501fc64: ntdll!_except_handler4+0 (777b99fa)
  CRT scope  0, filter: ntdll!__RtlUserThreadStart+3b (77827f8d)
                func:   ntdll!__RtlUserThreadStart+70 (77827fc7)
Invalid exception stack at ffffffff
0785f7ac: kernel32!_except_handler4+0 (7626fd89)
  CRT scope  1, func:   kernel32!WaitForSingleObjectEx+fc (762937c7)
  CRT scope  0, func:   kernel32!WaitForSingleObjectEx+110 (762937e2)
0785f840: ntdll!_except_handler4+0 (777b99fa)
  CRT scope  0, filter: ntdll!__RtlUserThreadStart+3b (77827f8d)
                func:   ntdll!__RtlUserThreadStart+70 (77827fc7)
Invalid exception stack at ffffffff
0868fcdc: kernel32!_except_handler4+0 (7626fd89)
  CRT scope  1, func:   kernel32!WaitForSingleObjectEx+fc (762937c7)
  CRT scope  0, func:   kernel32!WaitForSingleObjectEx+110 (762937e2)
0868fd74: ntdll!_except_handler4+0 (777b99fa)
```

```
  CRT scope  0, filter: ntdll!__RtlUserThreadStart+3b (77827f8d)
                func:   ntdll!__RtlUserThreadStart+70 (77827fc7)
Invalid exception stack at ffffffff
0b99fc58: ntdll!_except_handler4+0 (777b99fa)
  CRT scope  0, filter: ntdll!__RtlUserThreadStart+3b (77827f8d)
                func:   ntdll!__RtlUserThreadStart+70 (77827fc7)
Invalid exception stack at ffffffff
0bc7fab0: kernel32!_except_handler4+0 (7626fd89)
  CRT scope  1, func:   kernel32!WaitForMultipleObjectsEx+18a (7627a628)
  CRT scope  0, func:   kernel32!WaitForMultipleObjectsEx+186 (7627a630)
0bc7fb40: ntdll!_except_handler4+0 (777b99fa)
  CRT scope  0, filter: ntdll!__RtlUserThreadStart+3b (77827f8d)
                func:   ntdll!__RtlUserThreadStart+70 (77827fc7)
Invalid exception stack at ffffffff
0b04fc90: kernel32!_except_handler4+0 (7626fd89)
  CRT scope  1, func:   kernel32!WaitForMultipleObjectsEx+18a (7627a628)
  CRT scope  0, func:   kernel32!WaitForMultipleObjectsEx+186 (7627a630)
0b04fd98: ntdll!_except_handler4+0 (777b99fa)
  CRT scope  0, filter: ntdll!__RtlUserThreadStart+3b (77827f8d)
                func:   ntdll!__RtlUserThreadStart+70 (77827fc7)
Invalid exception stack at ffffffff
0bb7fc5c: ntdll!_except_handler4+0 (777b99fa)
  CRT scope  0, filter: ntdll!__RtlUserThreadStart+3b (77827f8d)
                func:   ntdll!__RtlUserThreadStart+70 (77827fc7)
Invalid exception stack at ffffffff
```

Note that here we look at anything abnormal such as raw “moduless” pointers. None found.

15. Close the log file:

```
0:004> .logclose
Closing open log file C:\AWMA-Dumps\M3.log
```



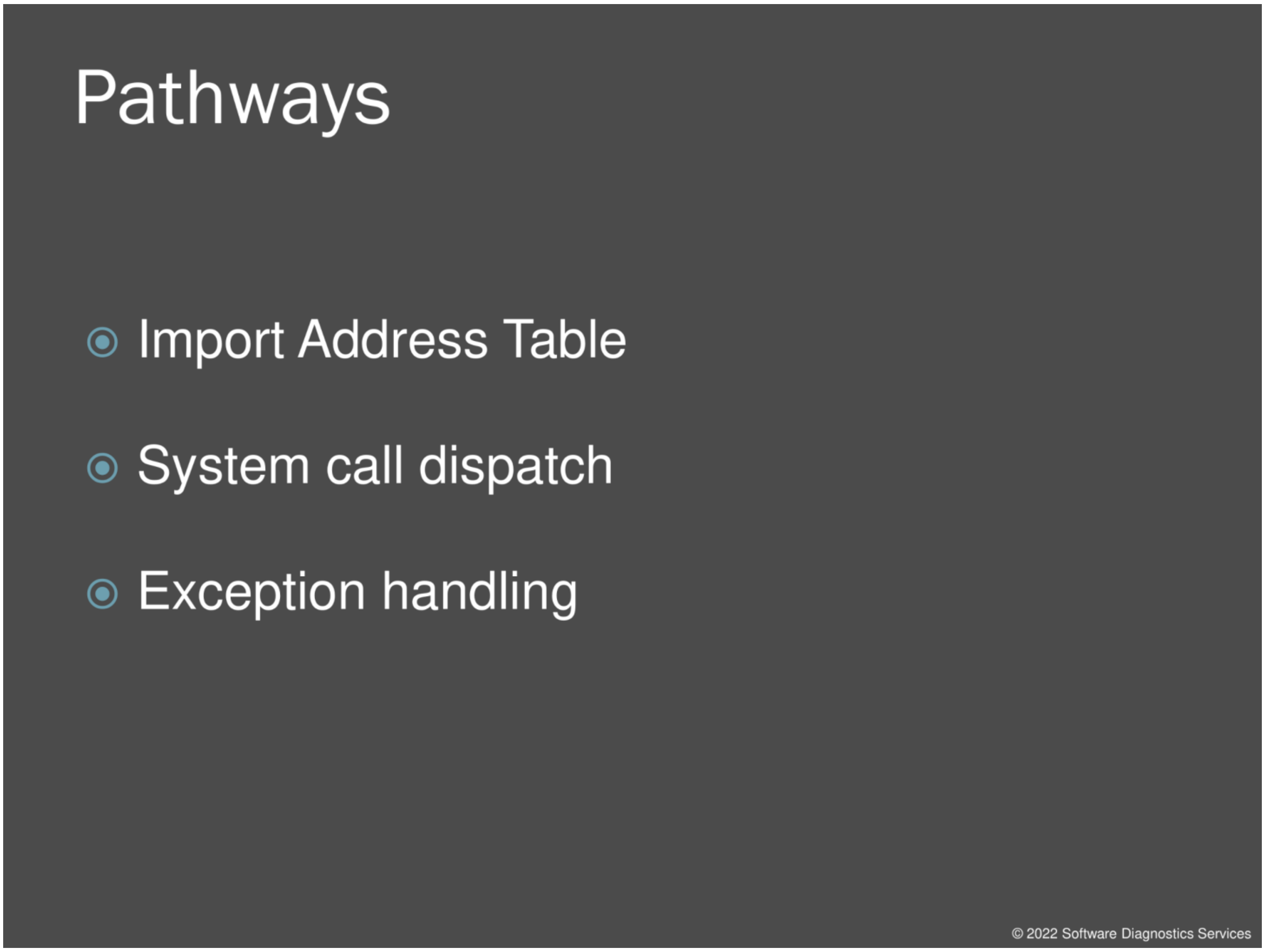
DLL Injection

Debugging TV Frame 0x20

Homework: InjectionResidue.DMP

We don’t cover DLL injection via remote threads and its possible execution residue in this training because a free case study is available. However, we provide you with a crash dump for homework so you can follow the presentation.

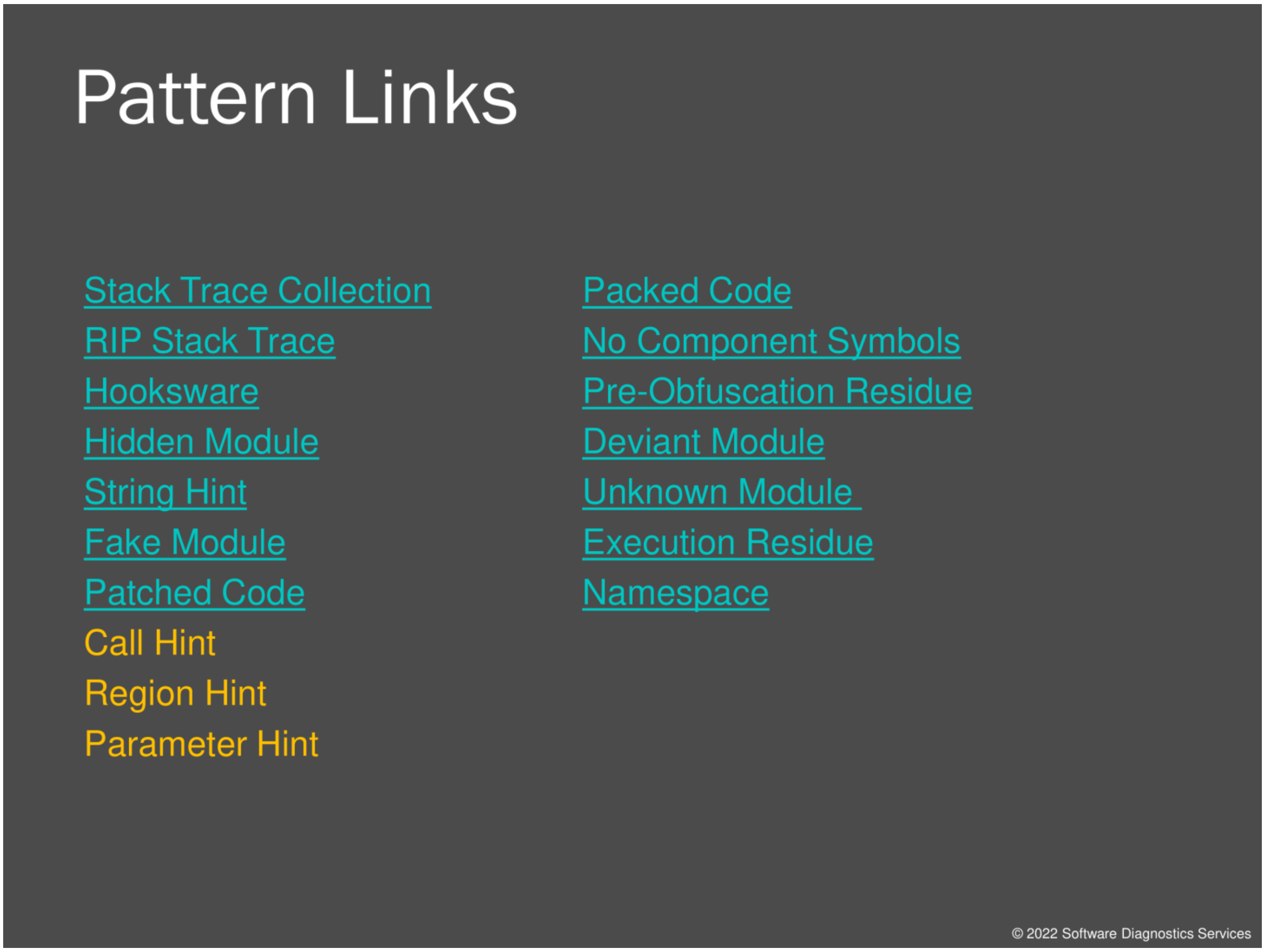
Debugging TV: http://www.debugging.tv/



Pathways

- Import Address Table
- System call dispatch
- Exception handling

To summarize, in exercise M3, we have seen 3 basic ways to drive malware execution: by hooking the Import Address Table functions, patching the system call dispatch mechanism, and by modifying exception handling chains and tables that deal with exception propagation.



Pattern Links

Stack Trace Collection
RIP Stack Trace
Hooksware
Hidden Module
String Hint
Fake Module
Patched Code
Call Hint
Region Hint
Parameter Hint

Packed Code
No Component Symbols
Pre-Obfuscation Residue
Deviant Module
Unknown Module
Execution Residue
Namespace

Here are links to descriptions of patterns we found in our examples (also available in Memory Dump Analysis Anthology, Encyclopedia of Crash Dump Analysis Patterns, and in this book Appendix):

Stack Trace Collection
https://www.dumpanalysis.org/blog/index.php/2007/09/14/crash-dump-analysis-patterns-part-27/

Packed Code
https://www.dumpanalysis.org/blog/index.php/2013/01/19/malware-analysis-patterns-part-3/

RIP Stack Trace
https://www.dumpanalysis.org/blog/index.php/2013/01/20/malware-analysis-patterns-part-11/

No Component Symbols
https://www.dumpanalysis.org/blog/index.php/2007/04/20/crash-dump-analysis-patterns-part-12/

Hooksware
https://www.dumpanalysis.org/blog/index.php/2008/08/10/hooksware/

Pre-Obfuscation Residue
https://www.dumpanalysis.org/blog/index.php/2013/01/19/malware-analysis-patterns-part-4/

Hidden Module
https://www.dumpanalysis.org/blog/index.php/2008/08/07/crash-dump-analysis-patterns-part-75/

Deviant Module
https://www.dumpanalysis.org/blog/index.php/2012/07/15/crash-dump-analysis-patterns-part-179/

String Hint
https://www.dumpanalysis.org/blog/index.php/2013/02/01/malware-analysis-patterns-part-18/

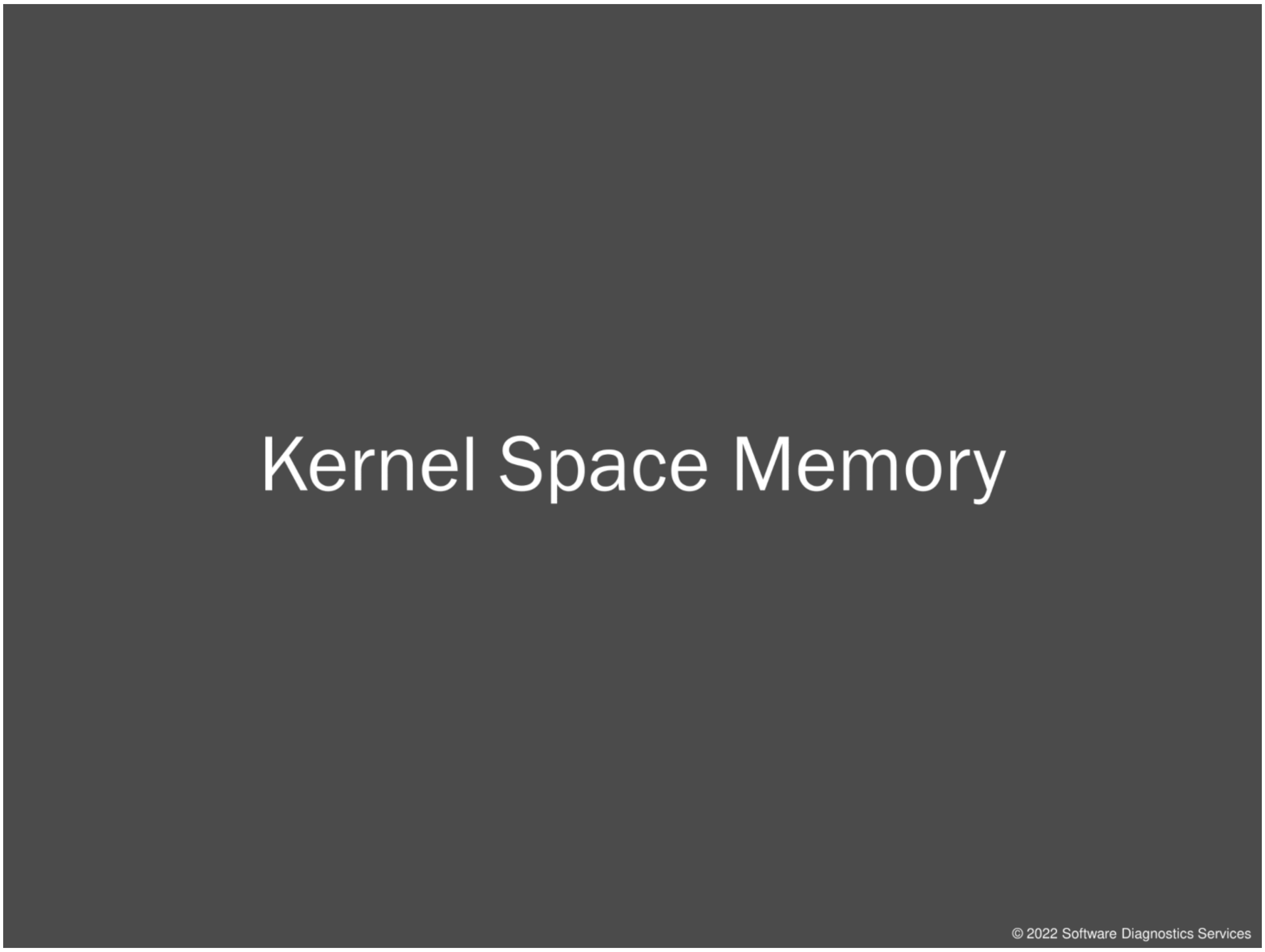
Unknown Module
https://www.dumpanalysis.org/blog/index.php/2007/08/16/crash-dump-analysis-patterns-part-22/

Fake Module
https://www.dumpanalysis.org/blog/index.php/2012/12/29/malware-analysis-patterns-part-2/

Execution Residue
https://www.dumpanalysis.org/blog/index.php/2008/04/29/crash-dump-analysis-patterns-part-60/

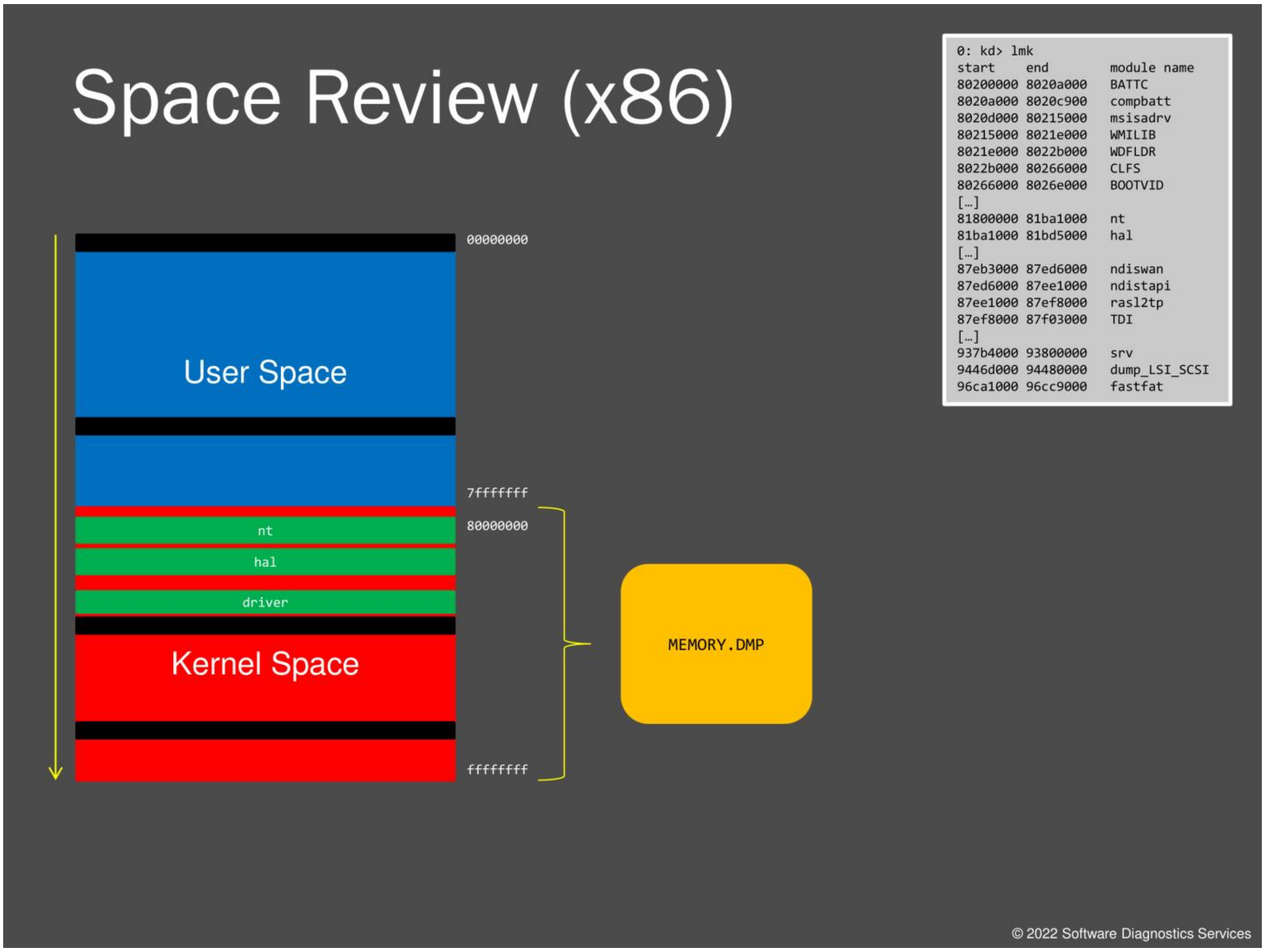
Patched Code
https://www.dumpanalysis.org/blog/index.php/2013/02/09/malware-analysis-patterns-part-21/

Namespace
https://www.dumpanalysis.org/blog/index.php/2013/02/05/malware-analysis-patterns-part-20/



Kernel Space Memory

Now we come to kernel space. Our goal is to show important commands and how their output helps in recognizing patterns of malware in the case of detected abnormal software behavior. All complete memory dumps were saved from virtualized 32-bit Windows Vista system, 64-bit Windows 8 system running on real hardware, and virtualized 64-bit Windows 11 system.

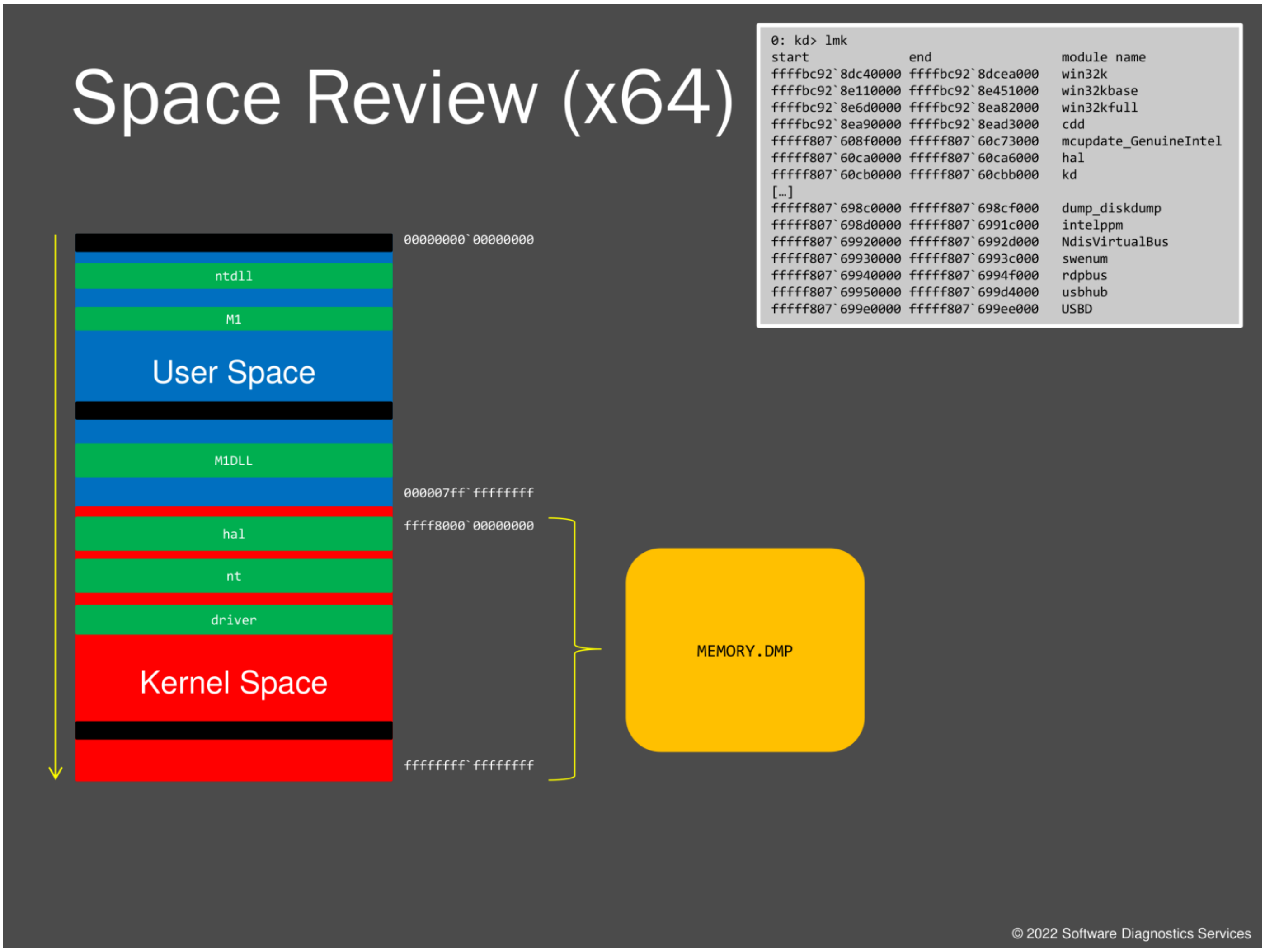


Space Review (x86)

```
0: kd> lmk
start    end         module name
80200000 8020a000   BATTC
8020a000 8020c900   compbatt
8020d000 80215000   msisadrv
80215000 8021e000   WMILIB
8021e000 8022b000   WDFLDR
8022b000 80266000   CLFS
80266000 8026e000   BOOTVID
[…]
81800000 81ba1000   nt
81ba1000 81bd5000   hal
[…]
87eb3000 87ed6000   ndiswan
87ed6000 87ee1000   ndistapi
87ee1000 87ef8000   rasl2tp
87ef8000 87f03000   TDI
[…]
937b4000 93800000   srv
9446d000 94480000   dump_LSI_SCSI
96ca1000 96cc9000   fastfat
```

© 2022 Software Diagnostics Services

Similar to a user space slide, I just briefly repeat that when the operating system is booted, its executable file is loaded into memory together with additional modules such as **hal.** This OS executable file can be found as **nt** module. During the driver loading stage, they are loaded dynamically like DLLs, and if they reference other DLLs, they are loaded too. Everything we learned about the PE header format is applicable here. In fact, .SYS file can be viewed as a system DLL, so there is no mystery there. There may be gaps between modules and other space regions like black regions in this picture. Some memory is also allocated for additional working regions needed for system execution. Kernel space usually has a 2 GB range, and we see addresses where modules are loaded by using the **lm** or **lmk** command. When we save a dump, all accessible memory, including loaded drivers, is saved. The dump is usually much smaller than 2 GB unless we have a kernel memory leak or some drivers are memory demanding.

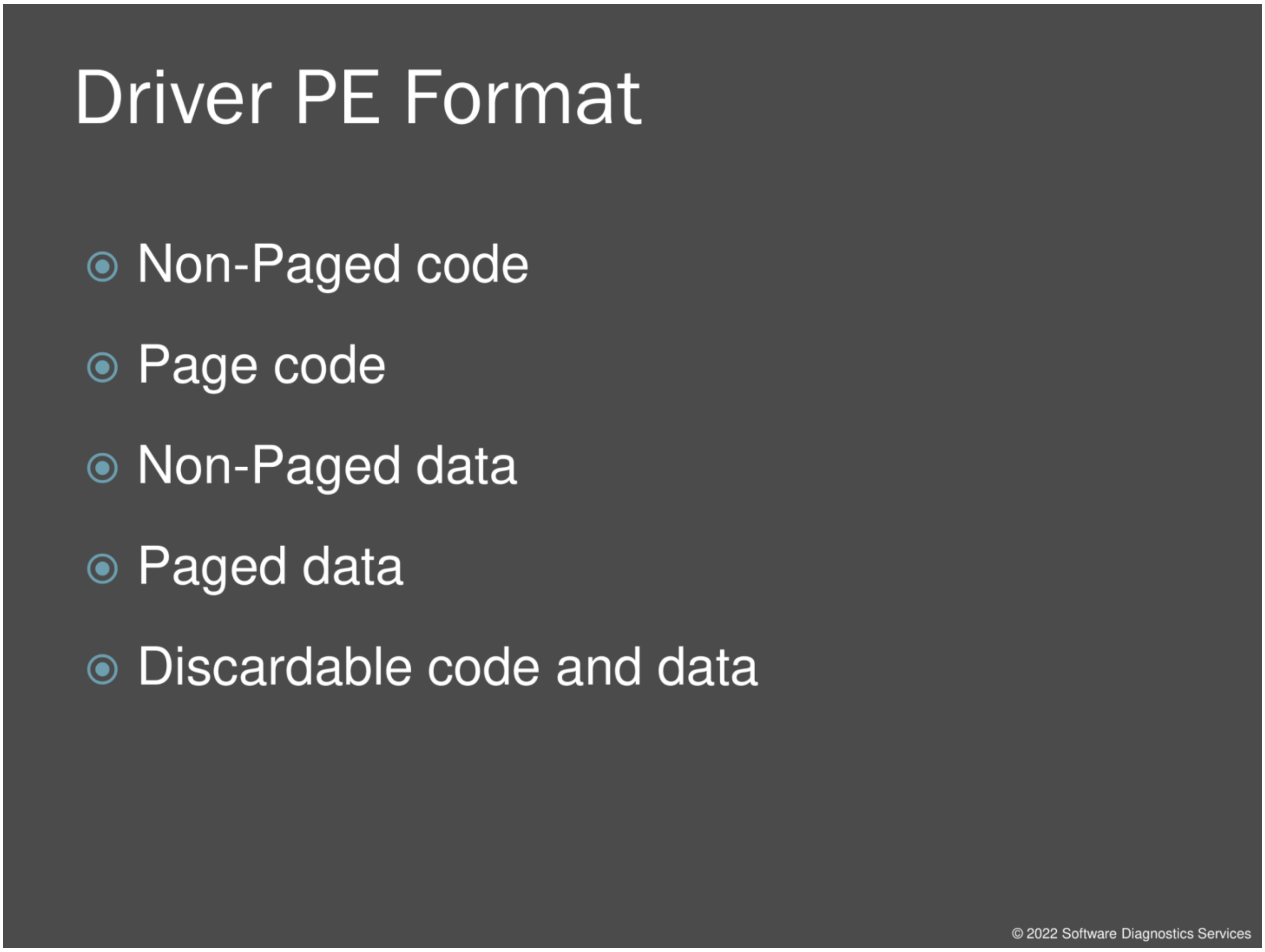


Space Review (x64)

```
0: kd> lmk
start             end                 module name
ffffbc92`8dc40000 ffffbc92`8dcea000   win32k
ffffbc92`8e110000 ffffbc92`8e451000   win32kbase
ffffbc92`8e6d0000 ffffbc92`8ea82000   win32kfull
ffffbc92`8ea90000 ffffbc92`8ead3000   cdd
fffff807`608f0000 fffff807`60c73000   mcupdate_GenuineIntel
fffff807`60ca0000 fffff807`60ca6000   hal
fffff807`60cb0000 fffff807`60cbb000   kd
[…]
fffff807`698c0000 fffff807`698cf000   dump_diskdump
fffff807`698d0000 fffff807`6991c000   intelppm
fffff807`69920000 fffff807`6992d000   NdisVirtualBus
fffff807`69930000 fffff807`6993c000   swenum
fffff807`69940000 fffff807`6994f000   rdpbus
fffff807`69950000 fffff807`699d4000   usbhub
fffff807`699e0000 fffff807`699ee000   USBD
```

© 2022 Software Diagnostics Services

Here we provide a picture of process space in 64-bit Windows. You see that kernel space is no longer restricted to 2 or 1 GB. We see that space distribution when we do an exercise. We now look at a typical driver PE header to see a few differences compared to user space modules.

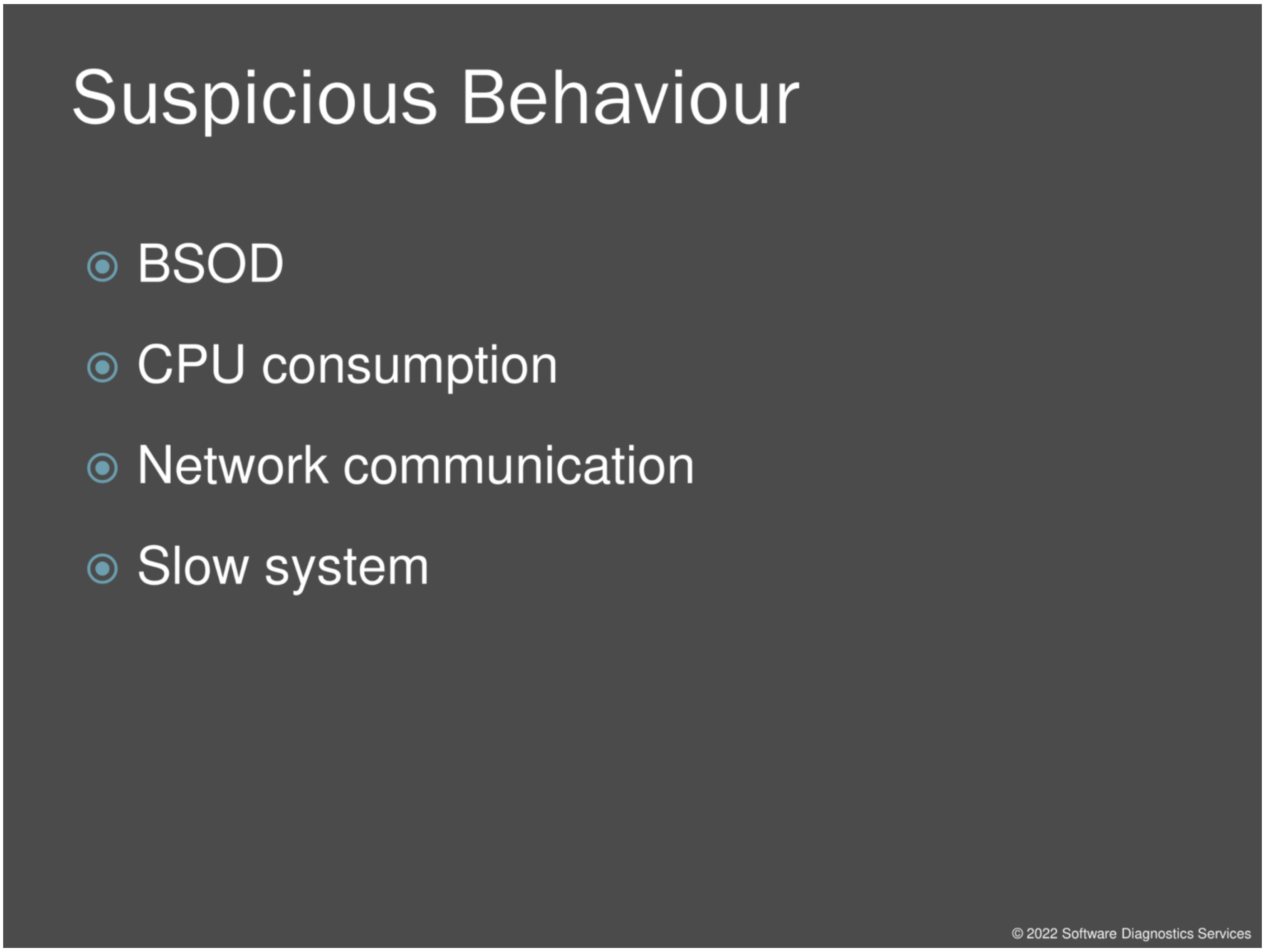


Driver PE Format

- Non-Paged code
- Page code
- Non-Paged data
- Paged data
- Discardable code and data

© 2022 Software Diagnostics Services

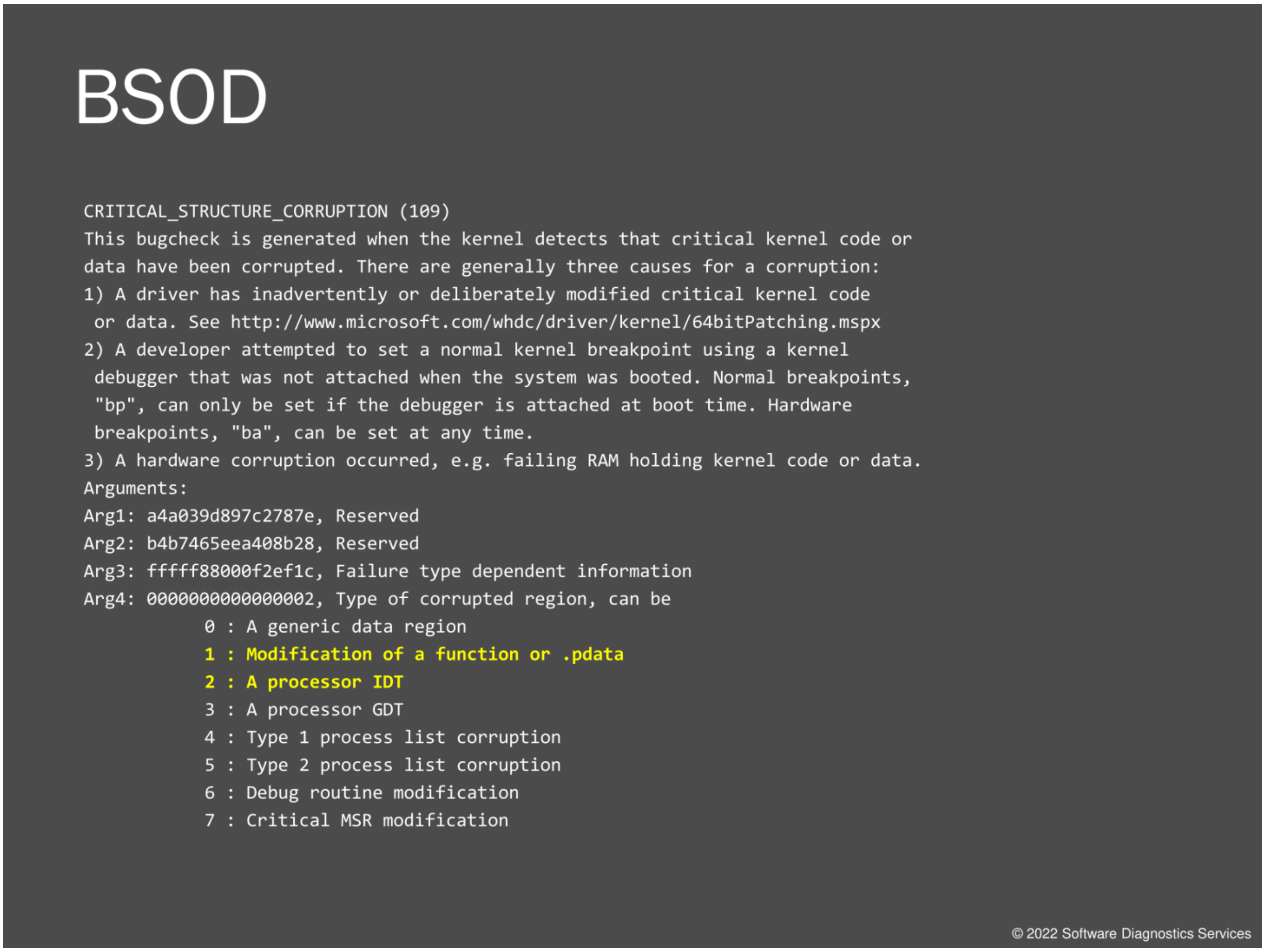
In user space executable files and dynamic link libraries, we saw one section for code and one for data. In kernel space, some code and data need to be always present in physical memory, and their sections are declared non-pageable. We also have sections for pageable code and data and also for discardable driver initialization code. All the rest is the same, including Import Address Tables.



Suspicious Behaviour

- BSOD
- CPU consumption
- Network communication
- Slow system

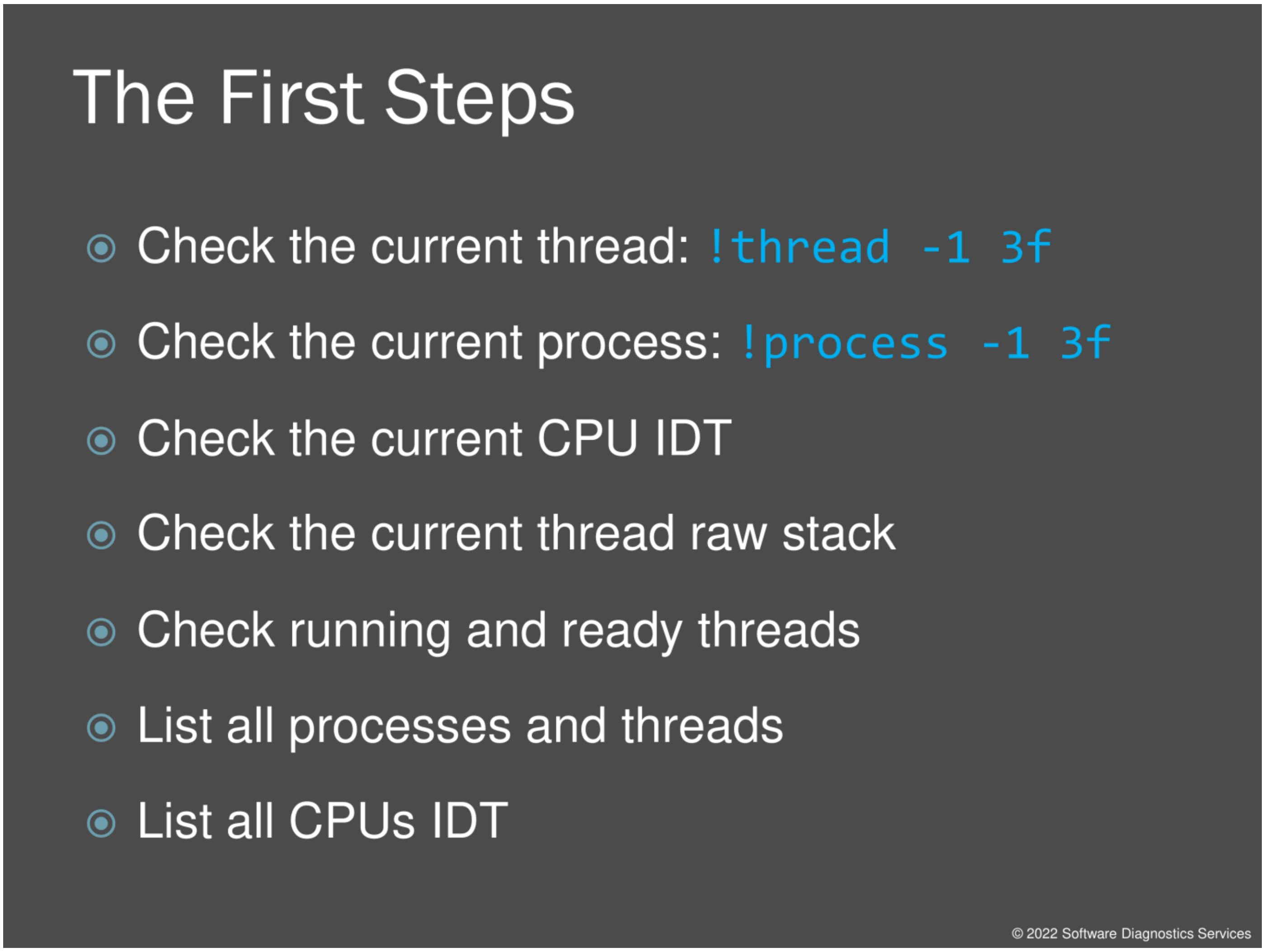
There are several cases of suspicious and abnormal system behavior that could have been potentially caused by malware or defective malware. For example, similar to heap corruption, a kernel-level rootkit could corrupt a kernel pool causing a blue screen with a corresponding bugcheck.



BSOD

```
CRITICAL_STRUCTURE_CORRUPTION (109)
This bugcheck is generated when the kernel detects that critical kernel code or
data have been corrupted. There are generally three causes for a corruption:
1) A driver has inadvertently or deliberately modified critical kernel code
 or data. See http://www.microsoft.com/whdc/driver/kernel/64bitPatching.mspx
2) A developer attempted to set a normal kernel breakpoint using a kernel
 debugger that was not attached when the system was booted. Normal breakpoints,
 "bp", can only be set if the debugger is attached at boot time. Hardware
 breakpoints, "ba", can be set at any time.
3) A hardware corruption occurred, e.g. failing RAM holding kernel code or data.
Arguments:
Arg1: a4a039d897c2787e, Reserved
Arg2: b4b7465eea408b28, Reserved
Arg3: fffff88000f2ef1c, Failure type dependent information
Arg4: 0000000000000002, Type of corrupted region, can be
            0 : A generic data region
            1 : Modification of a function or .pdata
            2 : A processor IDT
            3 : A processor GDT
            4 : Type 1 process list corruption
            5 : Type 2 process list corruption
            6 : Debug routine modification
            7 : Critical MSR modification
```

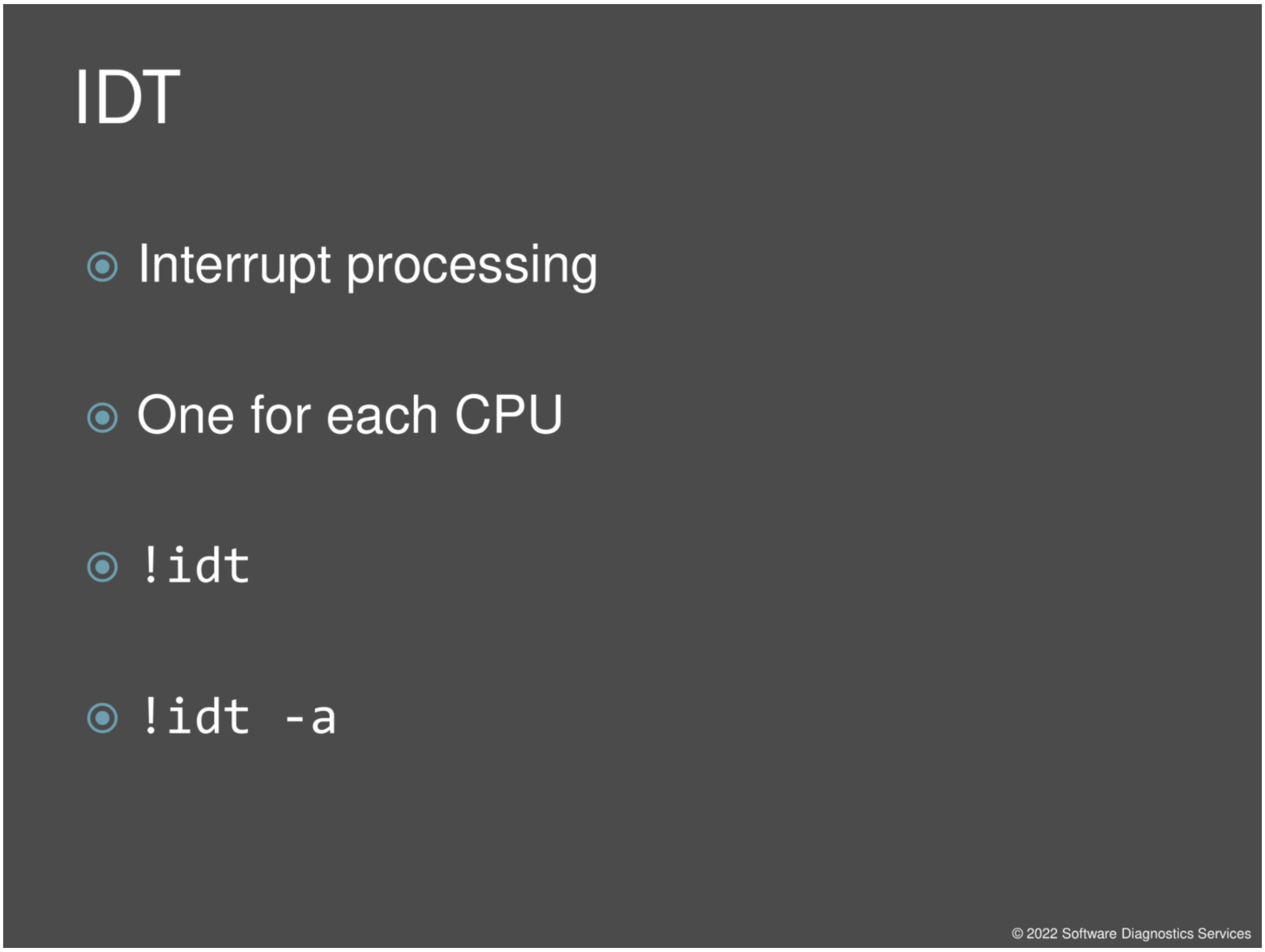
The latest Windows OS versions detect kernel structure modifications such as patching and, when detected, trigger a bugcheck. An example you see on this slide (the output from the **!analyze -v** command). Here a modification of IDT (Interrupt Descriptor Table) was detected. We cover IDT later in the next exercise.



The First Steps

- Check the current thread: `!thread -1 3f`
- Check the current process: `!process -1 3f`
- Check the current CPU IDT
- Check the current thread raw stack
- Check running and ready threads
- List all processes and threads
- List all CPUs IDT

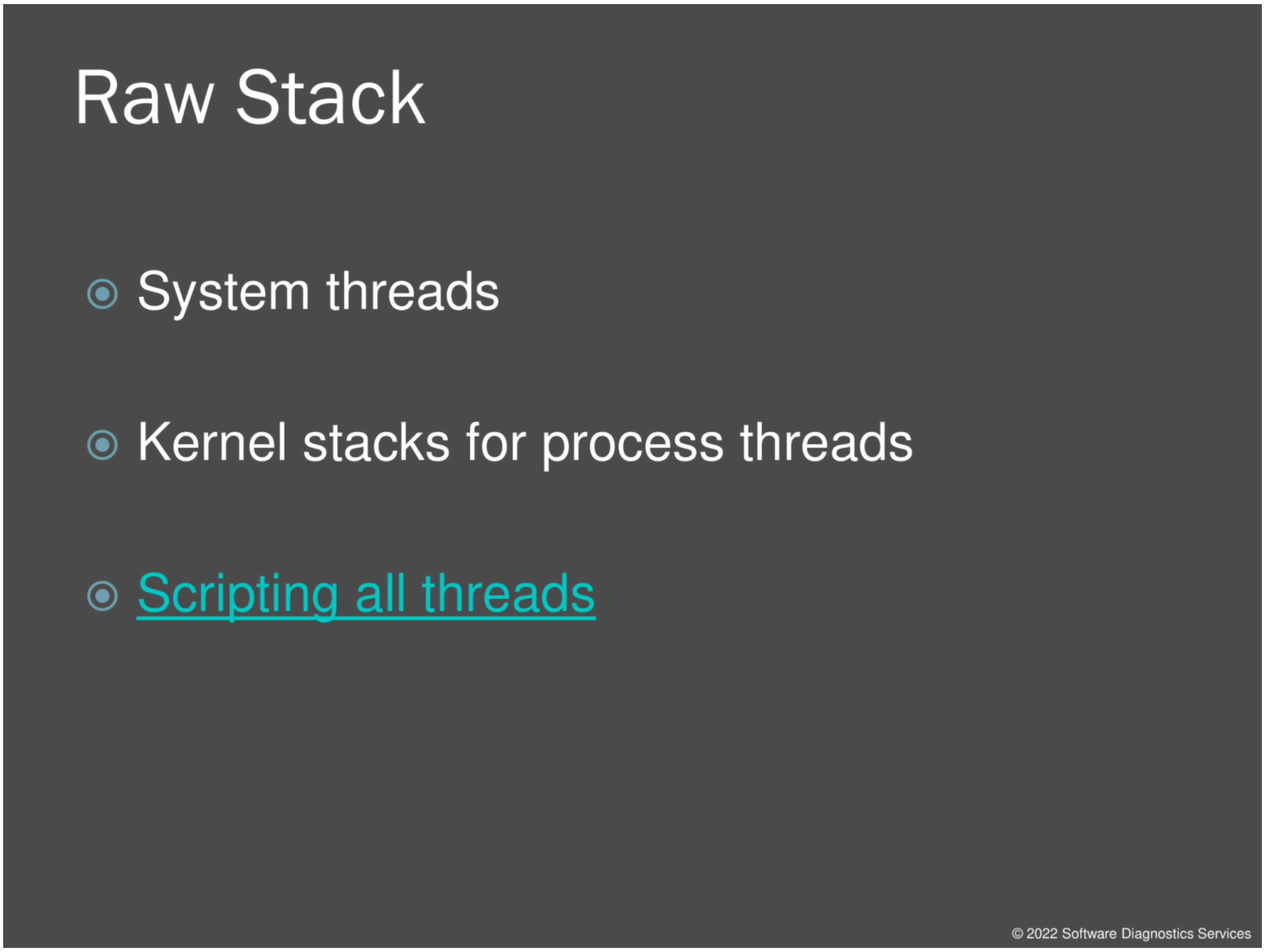
What are the first steps? In the case of BSOD, we might want to check the current thread and then the current process and CPU. The *3f* flag is needed for physical memory dump analysis, and it is good to learn it from the beginning as it has the same output for kernel space, even for just kernel memory dumps. Depending on the problem, we might also want to check running and ready for execution threads and also all processes and their threads. When looking at thread output, we might want to check kernel and user times spent, modules on stack traces, and the presence of any raw addresses. For CPUs, we might want to check their interrupt descriptor tables.



IDT

- Interrupt processing
- One for each CPU
- `!idt`
- `!idt -a`

IDT or Interrupt Descriptor Table is used to transfer execution to kernel functions upon an interrupt. Each entry in that table corresponds to an interrupt number (0 to 255) and has an associated pointer to some kernel procedure. Typical interrupts include page fault, divide-by-zero, and also hardware interrupts. We see this command in our next exercise. Just to mention that we might also want to check all interrupt table entries for the presence of any suspicious pointers because normally unused interrupt entries may potentially be used for communication. Also, note that each CPU has its own IDT.

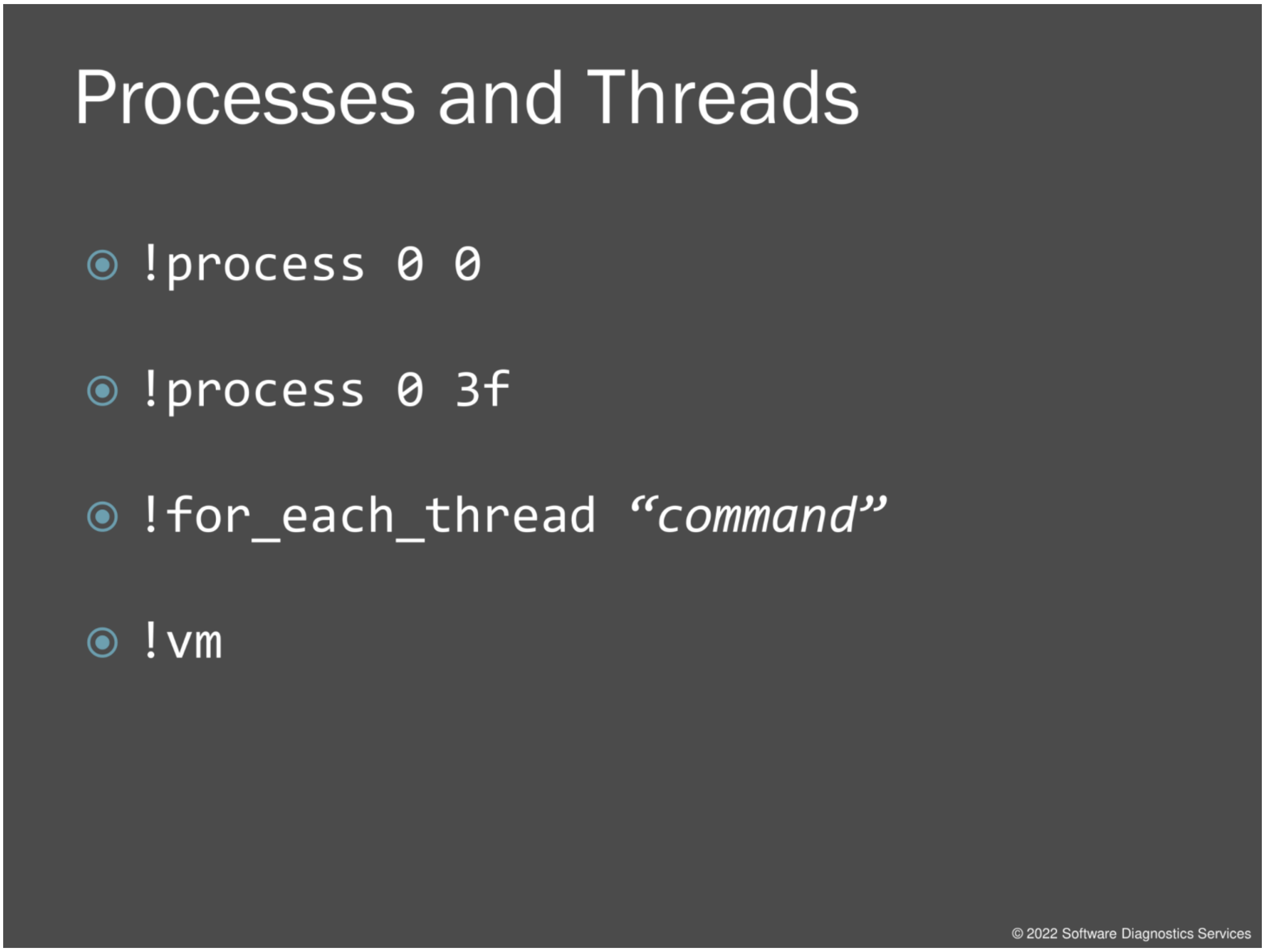


Raw Stack

- System threads
- Kernel stacks for process threads
- Scripting all threads

Please recall that we mentioned user space stack region in the previous exercises. The same region exists in the kernel for each thread, be it a system thread originated from the kernel or a thread originated from some process. In the latter case, we have 2 separate stack regions in different spaces.

Scripting all threads (also available in Volume 7 of Memory Dump Analysis Anthology and this book Appendix): https://www.dumpanalysis.org/blog/index.php/2012/01/22/raw-stack-dump-of-all-threads-part-5/.

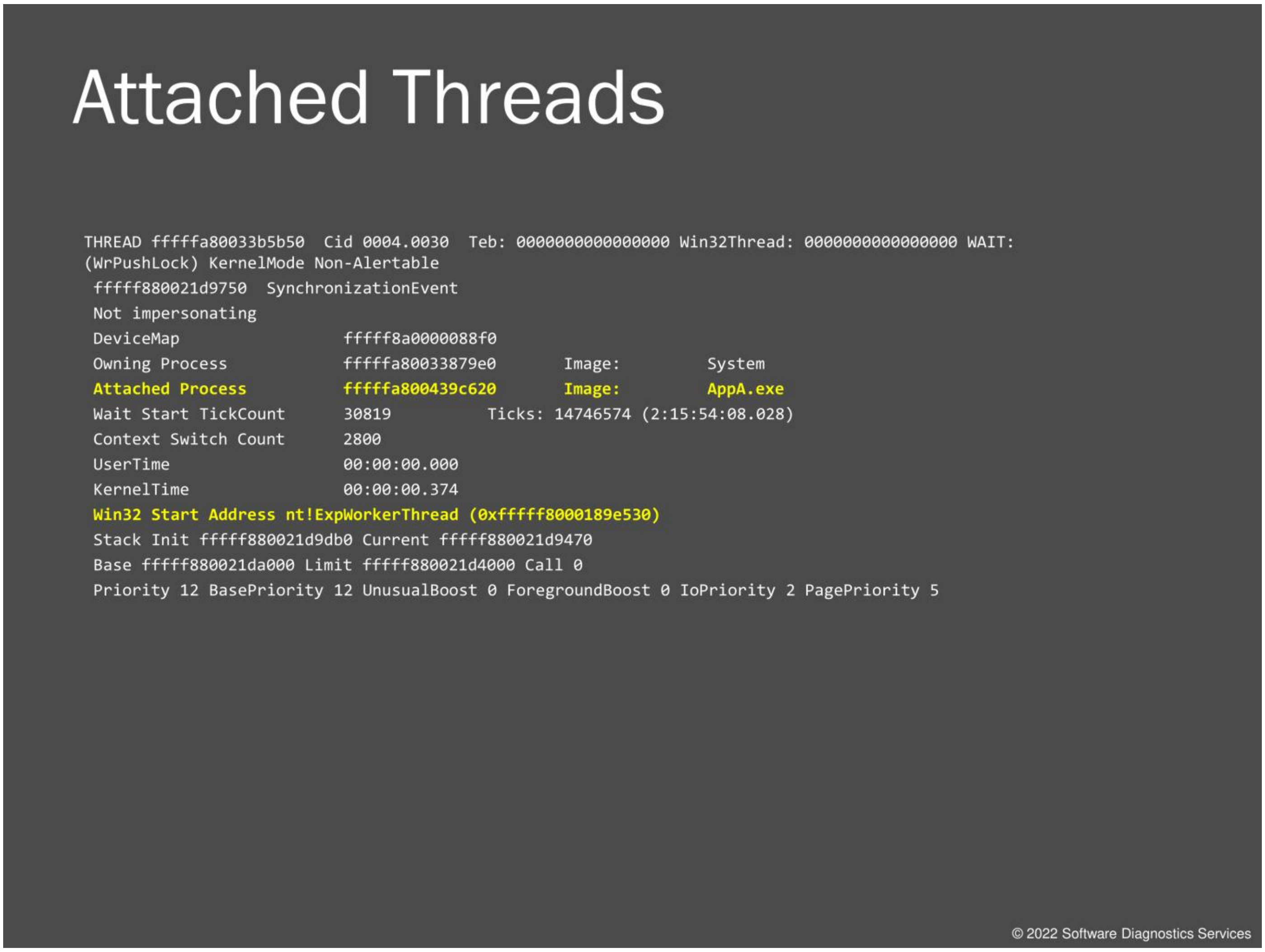


Processes and Threads

- !process 0 0
- !process 0 3f
- !for_each_thread *“command”*
- !vm

© 2022 Software Diagnostics Services

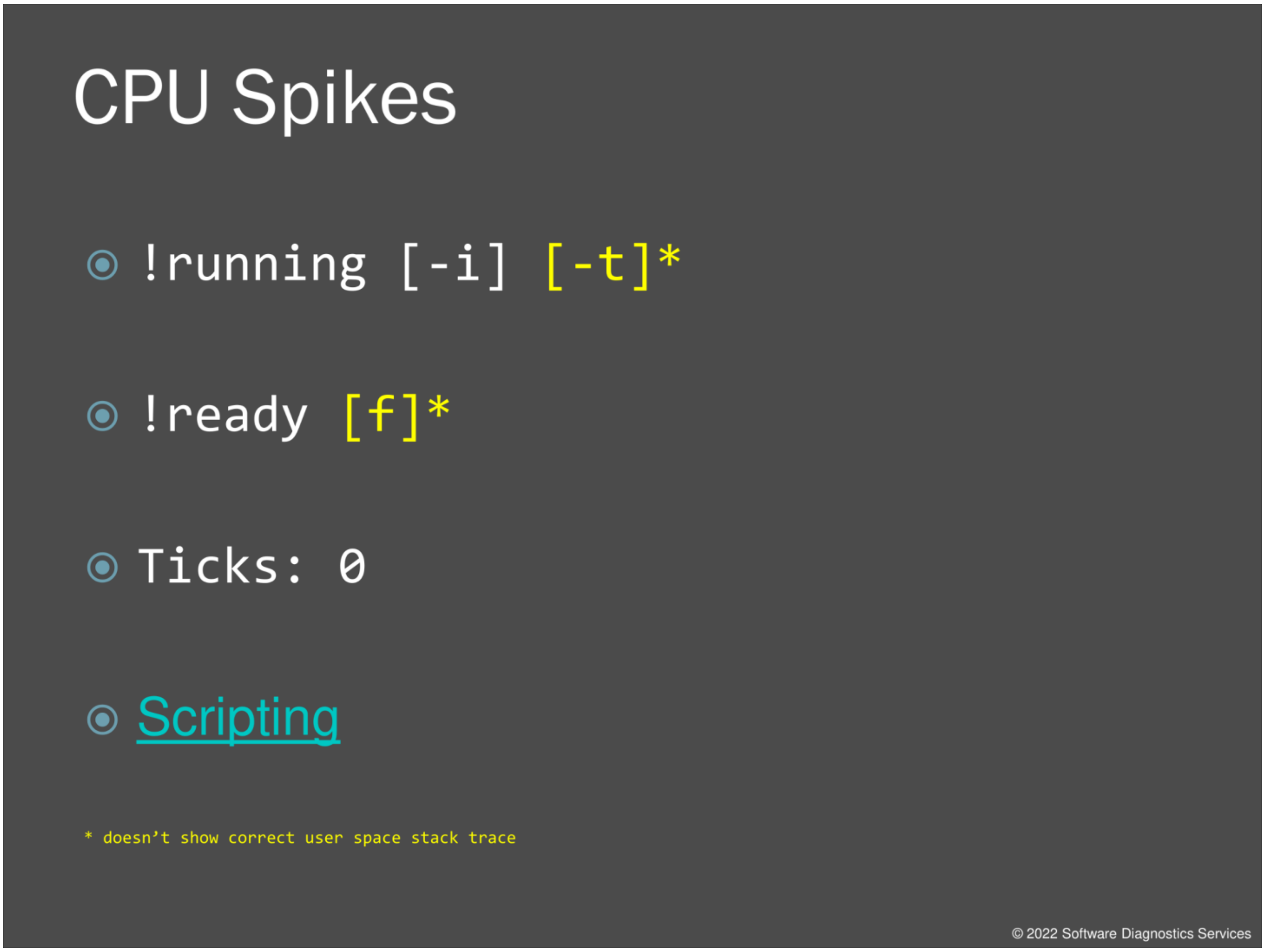
Obviously, the next thing we would like to check is processes and their thread stack traces. There are different ways to do it. The first 2 commands are similar to the individual thread and process commands, except that instead of -1, we put 0 to indicate all. And we can customize thread stack output with the 3rd command. An example is given in the previous slide scripting link. Process output is also available with the 4th command, where terminated but still referenced processes (the so-called “zombie processes”) are nicely grouped at the end of the output.



Attached Threads

```
THREAD fffffa80033b5b50  Cid 0004.0030  Teb: 0000000000000000 Win32Thread: 0000000000000000 WAIT:
(WrPushLock) KernelMode Non-Alertable
 fffff880021d9750  SynchronizationEvent
 Not impersonating
 DeviceMap                 fffff8a0000088f0
 Owning Process            fffffa80033879e0       Image:         System
 Attached Process          fffffa800439c620       Image:         AppA.exe
 Wait Start TickCount      30819          Ticks: 14746574 (2:15:54:08.028)
 Context Switch Count      2800
 UserTime                  00:00:00.000
 KernelTime                00:00:00.374
 Win32 Start Address nt!ExpWorkerThread (0xfffff8000189e530)
 Stack Init fffff880021d9db0 Current fffff880021d9470
 Base fffff880021da000 Limit fffff880021d4000 Call 0
 Priority 12 BasePriority 12 UnusualBoost 0 ForegroundBoost 0 IoPriority 2 PagePriority 5
```

Some system threads can be attached to a particular process if they need its resources. For example, on this fragment, we see the thread originated in kernel space but was attached to the AppA process, so it can access that process address space if needed.



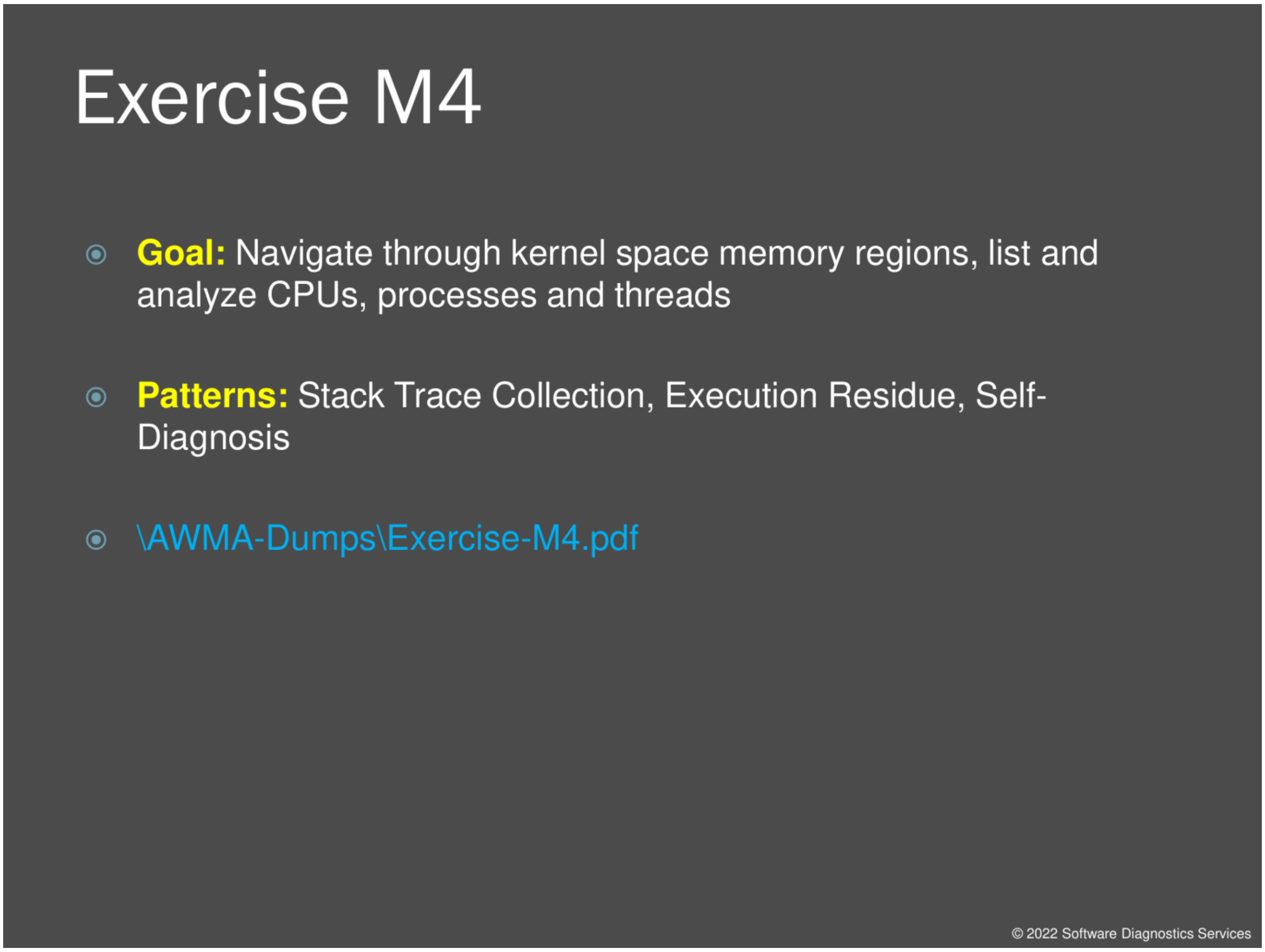
CPU Spikes

- !running [-i] [-t]*
- !ready [f]*
- Ticks: 0
- Scripting

* doesn't show correct user space stack trace

To check for CPU spiking activity and associated threads, we can use different commands. I also provided a link to WinDbg scripts that allow you to find out the most time-consuming thread in kernel and user modes in case it was consuming CPU sometime in the past, and this is not visible from the output of the first 2 commands or Ticks output.

Scripting CPU consumption (see also scripts in windbg.org and Volume 7 of Memory Dump Analysis Anthology, the full scripting case study is available in the Advanced Windows Memory Dump Analysis training course):
https://www.dumpanalysis.org/blog/index.php/2011/12/03/2-windbg-scripts-that-changed-the-world/.



Exercise M4

- **Goal:** Navigate through kernel space memory regions, list and analyze CPUs, processes and threads
- **Patterns:** Stack Trace Collection, Execution Residue, Self-Diagnosis
- \AWMA-Dumps\Exercise-M4.pdf

Now we analyze a complete memory dump but mainly focus on the kernel part for now.

Exercise M4

Goal: Navigate through kernel space memory regions, list and analyze CPUs, processes, and threads.

Patterns: Stack Trace Collection, Execution Residue, Self-Diagnosis.

1. Launch WinDbg Preview.

2. Open \AWMA-Dumps\Complete\MEMORY.DMP.

3. We get the dump file loaded:

```
Microsoft (R) Windows Debugger Version 10.0.25136.1001 AMD64
Copyright (c) Microsoft Corporation. All rights reserved.


Loading Dump File [C:\AWMA-Dumps\Complete\MEMORY.DMP]
Kernel Bitmap Dump File: Full address space is available


************* Path validation summary **************
Response                         Time (ms)     Location
Deferred                                       srv*
Symbol search path is: srv*
Executable search path is:
Windows 8 Kernel Version 9200 MP (2 procs) Free x64
Product: WinNt, suite: TerminalServer SingleUserTS
Edition build lab: 9200.16424.amd64fre.win8_gdr.120926-1855
Machine Name:
Kernel base = 0xfffff802`b3a89000 PsLoadedModuleList = 0xfffff802`b3d53a60
Debug session time: Tue Oct 30 21:22:24.413 2012 (UTC + 1:00)
System Uptime: 2 days 20:12:43.173
Loading Kernel Symbols
................................................................
.................................................................
........................
Loading User Symbols
.................................................................
............
Loading unloaded module list
...................................................................................
For analysis of this file, run !analyze -v
nt!KeBugCheckEx:
fffff802`b3b03d40 48894c2408      mov     qword ptr [rsp+8],rcx
ss:0018:fffff880`15925af0=00000000000000ef
```

4. Open a log file:

```
0: kd> .logopen C:\AWMA-Dumps\M4.log
Opened log file 'C:\AWMA-Dumps\M4.log'
```

5. How this dump was created is of no interest to us here so we skip **!analyze -v** step and look at kernel modules:

```
0: kd> lmk
start             end                 module name
fffff802`b309f000 fffff802`b30a8000   kd         (deferred)
fffff802`b3a1d000 fffff802`b3a89000   hal        (deferred)
fffff802`b3a89000 fffff802`b41d2000   nt         (pdb symbols)
C:\WinDbg.Docker.AWMA\mss\ntkrnlmp.pdb\9C419ACB04574E6D91857E85E46682032\ntkrnlmp.pdb
fffff880`00c00000 fffff880`00c7f000   CI         (deferred)
fffff880`00c7f000 fffff880`00ce2000   msrpc      (deferred)
fffff880`00cfd000 fffff880`00d5c000   mcupdate_GenuineIntel   (deferred)
fffff880`00d5c000 fffff880`00db8000   CLFS       (deferred)
fffff880`00db8000 fffff880`00ddb000   tm         (deferred)
fffff880`00ddb000 fffff880`00df0000   PSHED      (deferred)
fffff880`00df0000 fffff880`00dfa000   BOOTVID    (deferred)
fffff880`01000000 fffff880`0106d000   ACPI       (deferred)
fffff880`0106d000 fffff880`01077000   WMILIB     (deferred)
fffff880`01077000 fffff880`01081000   msisadrv   (deferred)
fffff880`010a8000 fffff880`0116a000   Wdf01000   (deferred)
fffff880`0116a000 fffff880`0117a000   WDFLDR     (deferred)
fffff880`0117a000 fffff880`01191000   acpiex     (deferred)
fffff880`01191000 fffff880`0119c000   WppRecorder   (deferred)
fffff880`0119c000 fffff880`011d9000   pci        (deferred)
fffff880`01200000 fffff880`01260000   volmgrx    (deferred)
fffff880`01264000 fffff880`012f0000   cng        (deferred)
fffff880`012f0000 fffff880`01318000   tpm        (deferred)
fffff880`01323000 fffff880`01330000   vdrvroot   (deferred)
fffff880`01330000 fffff880`01347000   pdc        (deferred)
fffff880`01347000 fffff880`01361000   partmgr    (deferred)
fffff880`01361000 fffff880`013aa000   spaceport   (deferred)
fffff880`013aa000 fffff880`013c2000   volmgr     (deferred)
fffff880`013c2000 fffff880`013cb000   intelide   (deferred)
fffff880`013cb000 fffff880`013da000   PCIIDEX    (deferred)
fffff880`01400000 fffff880`01456000   CLASSPNP   (deferred)
fffff880`01456000 fffff880`01465000   mouclass   (deferred)
fffff880`01465000 fffff880`0147c000   BTHUSB     (deferred)
fffff880`0148d000 fffff880`01516000   bxvbda     (deferred)
fffff880`01516000 fffff880`01576000   fltmgr     (deferred)
fffff880`01576000 fffff880`015b8000   WdFilter   (deferred)
fffff880`015b8000 fffff880`015c6000   TDI        (deferred)
fffff880`015c6000 fffff880`015f9580   usbvideo   (deferred)
fffff880`01600000 fffff880`01622000   tdx        (deferred)
fffff880`01622000 fffff880`0162e000   mouhid     (deferred)
fffff880`0162f000 fffff880`01969000   evbda      (deferred)
fffff880`01969000 fffff880`01983000   mountmgr   (deferred)
fffff880`01983000 fffff880`0198d000   atapi      (deferred)
fffff880`0198d000 fffff880`019c1000   ataport    (deferred)
fffff880`019c1000 fffff880`019db000   EhStorClass   (deferred)
fffff880`019db000 fffff880`019ef000   fileinfo   (deferred)
fffff880`019ef000 fffff880`019fc000   BasicRender   (deferred)
fffff880`01a00000 fffff880`01a2f000   ksecpkg    (deferred)
fffff880`01a2f000 fffff880`01a4b000   disk       (deferred)
fffff880`01a53000 fffff880`01c36000   Ntfs       (deferred)
fffff880`01c36000 fffff880`01c51000   ksecdd     (deferred)
fffff880`01c51000 fffff880`01c62000   pcw        (deferred)
fffff880`01c62000 fffff880`01c6c000   Fs_Rec     (deferred)
fffff880`01c6c000 fffff880`01d67000   ndis       (deferred)
fffff880`01d67000 fffff880`01dd7000   NETIO      (deferred)
```

```
fffff880`01df5000 fffff880`01dfd000   Beep       (deferred)
fffff880`01e00000 fffff880`01e3b000   rdyboost   (deferred)
fffff880`01e48000 fffff880`0207e000   tcpip      (deferred)
fffff880`0207e000 fffff880`020e6000   fwpkclnt   (deferred)
fffff880`020e6000 fffff880`02101000   wfplwfs    (deferred)
fffff880`02101000 fffff880`02177000   fvevol     (deferred)
fffff880`02177000 fffff880`021cc000   volsnap    (deferred)
fffff880`021cc000 fffff880`021e3000   mup        (deferred)
fffff880`021e3000 fffff880`021f7000   crashdmp   (deferred)
fffff880`021f7000 fffff880`02200000   Null       (deferred)
fffff880`03406000 fffff880`0356d000   dxgkrnl    (deferred)
fffff880`0356d000 fffff880`0357e000   watchdog   (deferred)
fffff880`0357e000 fffff880`035cc000   dxgmms1    (deferred)
fffff880`035cc000 fffff880`035dd000   BasicDisplay   (deferred)
fffff880`035dd000 fffff880`035ef000   Npfs       (deferred)
fffff880`035ef000 fffff880`035fb000   Msfs       (deferred)
fffff880`03600000 fffff880`0362a000   pacer      (deferred)
fffff880`0362a000 fffff880`03640000   vwififlt   (deferred)
fffff880`03640000 fffff880`03650000   netbios    (deferred)
fffff880`03650000 fffff880`036c2000   rdbss      (deferred)
fffff880`036c2000 fffff880`036ce000   BATTC      (deferred)
fffff880`036ce000 fffff880`036f1000   usbccgp    (deferred)
fffff880`036f1000 fffff880`03749000   netbt      (deferred)
fffff880`03749000 fffff880`037db000   afd        (deferred)
fffff880`037db000 fffff880`037e8000   kbdhid     (deferred)
fffff880`037e8000 fffff880`037f7000   kbdclass   (deferred)
fffff880`03800000 fffff880`0384b000   portcls    (deferred)
fffff880`0384d000 fffff880`038c8000   USBPORT    (deferred)
fffff880`038c8000 fffff880`038de000   usbehci    (deferred)
fffff880`038de000 fffff880`038f4000   HDAudBus   (deferred)
fffff880`038f4000 fffff880`03972000   usbhub     (deferred)
fffff880`03972000 fffff880`039ca000   HdAudio    (deferred)
fffff880`039ca000 fffff880`039d7000   hidusb     (deferred)
fffff880`039d7000 fffff880`039f2000   HIDCLASS   (deferred)
fffff880`039f2000 fffff880`039fa000   HIDPARSE   (deferred)
fffff880`03a00000 fffff880`03a0f000   CompositeBus   (deferred)
fffff880`03a0f000 fffff880`03a1a000   kdnic      (deferred)
fffff880`03a1a000 fffff880`03a2c000   umbus      (deferred)
fffff880`03a2c000 fffff880`03a48000   intelppm   (deferred)
fffff880`03a4c000 fffff880`03add000   csc        (deferred)
fffff880`03add000 fffff880`03af7000   wanarp     (deferred)
fffff880`03af7000 fffff880`03b05000   nsiproxy   (deferred)
fffff880`03b05000 fffff880`03b11000   npsvctrig   (deferred)
fffff880`03b11000 fffff880`03b1d000   mssmbios   (deferred)
fffff880`03b1d000 fffff880`03b2e000   discache   (deferred)
fffff880`03b2e000 fffff880`03b4f000   dfsc       (deferred)
fffff880`03b4f000 fffff880`03b55400   CmBatt     (deferred)
fffff880`03b5f000 fffff880`03b6b000   ndistapi   (deferred)
fffff880`03b6b000 fffff880`03b9a000   ndiswan    (deferred)
fffff880`03b9a000 fffff880`03bb8000   rassstp    (deferred)
fffff880`03bb8000 fffff880`03bd0000   AgileVpn   (deferred)
fffff880`03bd0000 fffff880`03bfc000   tunnel     (deferred)
fffff880`03e00000 fffff880`03e0e000   usbuhci    (deferred)
fffff880`03e17000 fffff880`043fee00   igdkmd64   (deferred)
fffff880`04400000 fffff880`04422000   bthpan     (deferred)
fffff880`04422000 fffff880`0443f000   hidbth     (deferred)
fffff880`0443f000 fffff880`0444c000   dump_dumpata   (deferred)
fffff880`0444c000 fffff880`04456000   dump_atapi   (deferred)
fffff880`04456000 fffff880`0446a000   dump_dumpfve   (deferred)
fffff880`0449c000 fffff880`045c0000   bthport    (deferred)
```

```
fffff880`045c0000 fffff880`045eb000   rfcomm     (deferred)
fffff880`045eb000 fffff880`045fd000   BthEnum    (deferred)
fffff880`04800000 fffff880`0480b000   rdpbus     (deferred)
fffff880`0480b000 fffff880`0481f000   NDProxy    (deferred)
fffff880`0481f000 fffff880`0482a000   USBD       (deferred)
fffff880`0482a000 fffff880`0484c000   drmk       (deferred)
fffff880`0484c000 fffff880`04851380   ksthunk    (deferred)
fffff880`04852000 fffff880`04d3f000   bcmwl63a   (deferred)
fffff880`04d3f000 fffff880`04d4c000   vwifibus   (deferred)
fffff880`04d4c000 fffff880`04d6d000   raspptp    (deferred)
fffff880`04d6d000 fffff880`04d92000   rasl2tp    (deferred)
fffff880`04d92000 fffff880`04dac000   raspppoe   (deferred)
fffff880`04dac000 fffff880`04dad480   swenum     (deferred)
fffff880`04dae000 fffff880`04dfd000   ks         (deferred)
fffff880`15262000 fffff880`1528a000   luafv      (deferred)
fffff880`1528a000 fffff880`1529e000   lltdio     (deferred)
fffff880`1529e000 fffff880`1530c000   nwifi      (deferred)
fffff880`1530c000 fffff880`15320000   ndisuio    (deferred)
fffff880`15320000 fffff880`15338000   rspndr     (deferred)
fffff880`15338000 fffff880`15342000   vwifimp    (deferred)
fffff880`15342000 fffff880`1535e000   Ndu        (deferred)
fffff880`1535e000 fffff880`153eb000   srv        (deferred)
fffff880`15a00000 fffff880`15a62000   mrxsmb     (deferred)
fffff880`15a62000 fffff880`15aad000   mrxsmb10   (deferred)
fffff880`15ab3000 fffff880`15b8f000   HTTP       (deferred)
fffff880`15b8f000 fffff880`15baf000   bowser     (deferred)
fffff880`15baf000 fffff880`15bc6000   mpsdrv     (deferred)
fffff880`15bc6000 fffff880`15c00000   mrxsmb20   (deferred)
fffff880`15c00000 fffff880`15ca0000   srv2       (deferred)
fffff880`15ca0000 fffff880`15cab000   rdpvideominiport   (deferred)
fffff880`15cae000 fffff880`15cbc000   monitor    (deferred)
fffff880`15cbc000 fffff880`15cc9000   condrv     (deferred)
fffff880`15ccd000 fffff880`15d98000   peauth     (deferred)
fffff880`15d98000 fffff880`15da3000   secdrv     (deferred)
fffff880`15da3000 fffff880`15de7000   srvnet     (deferred)
fffff880`15de7000 fffff880`15df9000   tcpipreg   (deferred)
fffff960`0007a000 fffff960`0046f000   win32k     (deferred)
fffff960`006d1000 fffff960`006da000   TSDDD      (deferred)
fffff960`008a4000 fffff960`008da000   cdd        (deferred)

Unloaded modules:
fffff880`153eb000 fffff880`153f8000   hiber_ataport.sys
fffff880`15200000 fffff880`1520a000   hiber_atapi.sys
fffff880`1520a000 fffff880`1521e000   hiber_dumpfve.sys
fffff880`15ca0000 fffff880`15ca8000   drmkaud.sys
fffff880`15dfc000 fffff880`15dfe000   MSTEE.sys
fffff880`15df9000 fffff880`15dfc000   MSKSSRV.sys
fffff880`15ccb000 fffff880`15ccd000   MSPQM.sys
fffff880`15cc9000 fffff880`15ccb000   MSPCLOCK.sys
fffff880`15ca0000 fffff880`15cae000   monitor.sys
fffff880`0446a000 fffff880`04478000   monitor.sys
fffff880`01e3b000 fffff880`01e48000   dump_ataport.sys
fffff880`01dd7000 fffff880`01de1000   dump_atapi.sys
fffff880`01de1000 fffff880`01df5000   dump_dumpfve.sys
fffff880`03b4f000 fffff880`03b5f000   dam.sys
fffff880`01456000 fffff880`01487000   cdrom.sys
fffff880`01318000 fffff880`01323000   WdBoot.sys
fffff880`021e3000 fffff880`021ef000   hwpolicy.sys
fffff880`00cf0000 fffff880`00cfd000   ApiSetSchema.dll
000007fe`eb670000 000007fe`eb682000   BROWCLI.DLL
```

```
000007fe`f48c0000 000007fe`f48e4000   srvcli.dll
000007fe`e6830000 000007fe`e68c5000   tiptsf.dll
000007fe`e7820000 000007fe`e7897000   verifier.dll
000007fe`f7b20000 000007fe`f7b27000   psapi.dll
000007fe`f0ca0000 000007fe`f0ca9000   version.dll
000007fe`eb1c0000 000007fe`eb237000   verifier.dll
000007fe`f7b20000 000007fe`f7b27000   psapi.dll
000007fe`f0ca0000 000007fe`f0ca9000   version.dll
000007fe`f4110000 000007fe`f4157000   AUTHZ.dll
000007fe`f1b70000 000007fe`f1b88000   slc.dll
000007fe`efcc0000 000007fe`efcd7000   MPR.dll
000007fe`ea520000 000007fe`ea619000   ACLUI.dll
000007fe`f3840000 000007fe`f3864000   NTDSAPI.dll
000007fe`f3790000 000007fe`f3799000   DSROLE.dll
000007fe`ec300000 000007fe`ec32e000   srmshell.dll
000007fe`f3800000 000007fe`f381d000   ATL.DLL
000007fe`ec2e0000 000007fe`ec2fb000   SrmTrace.DLL
000007fe`ec330000 000007fe`ec345000   cryptext.dll
000007fe`eb1a0000 000007fe`eb233000   CRYPTUI.dll
000007fe`ecb90000 000007fe`ecbc0000   syncui.dll
000007fe`ec350000 000007fe`ec36b000   SYNCENG.dll
000007fe`efc50000 000007fe`efc5b000   LINKINFO.dll
000007fe`f0f00000 000007fe`f0f0f000   acppage.dll
000007fe`ebf20000 000007fe`ebf23000   sfc.dll
000007fe`e8e20000 000007fe`e90dd000   msi.dll
000007fe`eef30000 000007fe`eef40000   sfc_os.DLL
000007fe`f4ec0000 000007fe`f4f15000   WINTRUST.DLL
000007fe`f7ce0000 000007fe`f7cf4000   imagehlp.dll
000007fe`f5100000 000007fe`f52d7000   CRYPT32.dll
000007fe`f4ea0000 000007fe`f4eb6000   MSASN1.dll
000007fe`f4870000 000007fe`f4897000   ncrypt.dll
000007fe`f4830000 000007fe`f4865000   NTASN1.dll
000007fe`e8620000 000007fe`e8773000   wdc.dll
000007fe`ea680000 000007fe`ea693000   pdhui.dll
000007fe`f7a20000 000007fe`f7ac1000   COMDLG32.dll
000007fe`e8560000 000007fe`e861e000   ODBC32.dll
000007fe`edf30000 000007fe`edf3b000   Secur32.dll
000007fe`f0ca0000 000007fe`f0ca9000   VERSION.dll
000007fe`e7740000 000007fe`e7893000   PLA.dll
000007fe`e8b30000 000007fe`e8b7c000   pdh.dll
000007fe`f3690000 000007fe`f3774000   tdh.dll
000007fe`ec170000 000007fe`ec195000   Cabinet.dll
000007fe`f3a50000 000007fe`f3abc000   wevtapi.dll
000007fe`ea440000 000007fe`ea457000   UTILDLL.dll
000007fe`f3820000 000007fe`f3835000   NETAPI32.dll
000007fe`f4440000 000007fe`f4474000   LOGONCLI.DLL
000007fe`eb670000 000007fe`eb682000   BROWCLI.DLL
000007fe`f48c0000 000007fe`f48e4000   srvcli.dll
000007fe`f4ba0000 000007fe`f4bcc000   SSPICLI.DLL
000007fe`e8620000 000007fe`e8773000   wdc.dll
000007fe`ea680000 000007fe`ea693000   pdhui.dll
000007fe`f7a20000 000007fe`f7ac1000   COMDLG32.dll
000007fe`e8560000 000007fe`e861e000   ODBC32.dll
000007fe`edf30000 000007fe`edf3b000   Secur32.dll
000007fe`f0ca0000 000007fe`f0ca9000   VERSION.dll
000007fe`e7740000 000007fe`e7893000   PLA.dll
000007fe`e8b30000 000007fe`e8b7c000   pdh.dll
000007fe`f3690000 000007fe`f3774000   tdh.dll
000007fe`ec170000 000007fe`ec195000   Cabinet.dll
000007fe`f3a50000 000007fe`f3abc000   wevtapi.dll
```

```
000007fe`ea440000 000007fe`ea457000   UTILDLL.dll
000007fe`f3820000 000007fe`f3835000   NETAPI32.dll
000007fe`f4440000 000007fe`f4474000   LOGONCLI.DLL
```

Notice the unload modules list. These names can also be considered a part of execution residue.

6. Let’s check a typical driver module header and IAT:

```
0: kd> !dh disk

File Type: EXECUTABLE IMAGE
FILE HEADER VALUES
    8664 machine (X64)
       9 number of sections
5010AB85 time date stamp Thu Jul 26 03:29:25 2012

       0 file pointer to symbol table
       0 number of symbols
      F0 size of optional header
      22 characteristics
            Executable
            App can handle >2gb addresses

OPTIONAL HEADER VALUES
     20B magic #
   10.10 linker version
    EA00 size of code
    8200 size of initialized data
       0 size of uninitialized data
    215C address of entry point
    1000 base of code
         ----- new -----
fffff802b5567000 image base
    1000 section alignment
     200 file alignment
       1 subsystem (Native)
    6.02 operating system version
    6.02 image version
    6.02 subsystem version
   1C000 size of image
     400 size of headers
   24F95 checksum
0000000000040000 size of stack reserve
0000000000001000 size of stack commit
0000000000100000 size of heap reserve
0000000000001000 size of heap commit
       0  DLL characteristics
       0 [       0] address [size] of Export Directory
   15118 [      3C] address [size] of Import Directory
   16000 [    4258] address [size] of Resource Directory
    A000 [     EAC] address [size] of Exception Directory
   17000 [    20F0] address [size] of Security Directory
   1B000 [      A0] address [size] of Base Relocation Directory
    5A54 [      38] address [size] of Debug Directory
       0 [       0] address [size] of Description Directory
       0 [       0] address [size] of Special Directory
       0 [       0] address [size] of Thread Storage Directory
    6810 [      70] address [size] of Load Configuration Directory
       0 [       0] address [size] of Bound Import Directory
    6000 [     2D8] address [size] of Import Address Table Directory
       0 [       0] address [size] of Delay Import Directory
       0 [       0] address [size] of COR20 Header Directory
       0 [       0] address [size] of Reserved Directory


SECTION HEADER #1
   .text name
    4AB5 virtual size
    1000 virtual address
    4C00 size of raw data
     400 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
68000020 flags
         Code
         Not Paged
         (no align specified)
         Execute Read

Debug Directories(2)
       Type       Size     Address  Pointer
       cv           21        5a94     4e94   Format: RSDS, guid, 2, disk.pdb
       (    10)       8         5a8c     4e8c

SECTION HEADER #2
  .rdata name
    2270 virtual size
    6000 virtual address
    2400 size of raw data
    5000 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
48000040 flags
         Initialized Data
         Not Paged
         (no align specified)
         Read Only

SECTION HEADER #3
   .data name
     2C5 virtual size
    9000 virtual address
     400 size of raw data
    7400 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
C8000040 flags
         Initialized Data
         Not Paged
         (no align specified)
         Read Write

SECTION HEADER #4
```

```
  .pdata name
     EAC virtual size
    A000 virtual address
    1000 size of raw data
    7800 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
48000040 flags
         Initialized Data
         Not Paged
         (no align specified)
         Read Only

SECTION HEADER #5
    PAGE name
    7E59 virtual size
    B000 virtual address
    8000 size of raw data
    8800 file pointer to raw data
 1A3A000 file pointer to relocation table
FFFFF880 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
60000020 flags
         Code
         (no align specified)
         Execute Read

SECTION HEADER #6
    PAGE name
     2A0 virtual size
   13000 virtual address
     400 size of raw data
   10800 file pointer to raw data
 1A42000 file pointer to relocation table
FFFFF880 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
C0000040 flags
         Initialized Data
         (no align specified)
         Read Write

SECTION HEADER #7
    INIT name
    1C9C virtual size
   14000 virtual address
    1E00 size of raw data
   10C00 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
E2000020 flags
         Code
         Discardable
         (no align specified)
         Execute Read Write
```

```
SECTION HEADER #8
   .rsrc name
    4258 virtual size
   16000 virtual address
    4400 size of raw data
   12A00 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
42000040 flags
         Initialized Data
         Discardable
         (no align specified)
         Read Only

SECTION HEADER #9
  .reloc name
      A0 virtual size
   1B000 virtual address
     200 size of raw data
   16E00 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
42000040 flags
         Initialized Data
         Discardable
         (no align specified)
         Read Only
```

Note different code and data sections for non-pageable, pageable, and discardable code and data. For the image base address, we need to take the value from the output of the **lm m** command:

```
0: kd> lm m disk
start             end                 module name
fffff880`01a2f000 fffff880`01a4b000   disk       (deferred)

0: kd> dps fffff880`01a2f000+6000 L2D8/8
fffff880`01a35000  fffff802`b3aeb4d0 nt!IoGetAttachedDeviceReference
fffff880`01a35008  fffff802`b3b8cc10 nt!IoAttachDeviceToDeviceStack
fffff880`01a35010  fffff802`b3b63b10 nt!IoAllocateIrp
fffff880`01a35018  fffff802`b3b2b120 nt!RtlCompareMemory
fffff880`01a35020  fffff802`b3af99a0 nt!ObfDereferenceObject
fffff880`01a35028  fffff802`b3aeb1f0 nt!IoQueueWorkItem
fffff880`01a35030  fffff802`b3b3c3b0 nt!IofCallDriver
fffff880`01a35038  fffff802`b3b3d1f0 nt!IoGetIoPriorityHint
fffff880`01a35040  fffff802`b3c48d7c nt!ExInterlockedPopEntryList
fffff880`01a35048  fffff802`b3b72a70 nt!MmBuildMdlForNonPagedPool
fffff880`01a35050  fffff802`b3b4d960 nt!IoFreeMdl
fffff880`01a35058  fffff802`b3b471e0 nt!IoFreeIrp
fffff880`01a35060  fffff802`b3c48e14 nt!ExInterlockedPushEntryList
fffff880`01a35068  fffff802`b3aef97c nt!ExInitializePushLock
fffff880`01a35070  fffff802`b3b29a50 nt!KeWaitForSingleObject
fffff880`01a35078  fffff802`b3f69f30 nt!IoReadDiskSignature
fffff880`01a35080  fffff802`b3b04be0 nt!ZwQueryValueKey
fffff880`01a35088  fffff802`b3ec3bac nt!RtlUnicodeStringToInteger
```

```
fffff880`01a35090  fffff802`b3b04b40 nt!ZwOpenKey
fffff880`01a35098  fffff802`b3f87600 nt!IoGetConfigurationInformation
fffff880`01a350a0  fffff802`b3f94cf0 nt!IoDeleteSymbolicLink
fffff880`01a350a8  fffff802`b3ac6f60 nt!KeInitializeMutex
fffff880`01a350b0  fffff802`b3a8c0a0 nt!HalExamineMBR
fffff880`01a350b8  fffff802`b3f5a0cc nt!RtlQueryRegistryValues
fffff880`01a350c0  fffff802`b3d70104 nt!InitSafeBootMode
fffff880`01a350c8  fffff802`b3b8148c nt!vsnprintf
fffff880`01a350d0  fffff802`b3f94c70 nt!IoCreateSymbolicLink
fffff880`01a350d8  fffff802`b3e1d280 nt!IoOpenDeviceRegistryKey
fffff880`01a350e0  fffff802`b3bac250 nt!IoSetActivityIdIrp
fffff880`01a350e8  fffff802`b3b04ae0 nt!ZwClose
fffff880`01a350f0  fffff802`b3af33cc nt!vsnwprintf
fffff880`01a350f8  fffff802`b3ab17dc nt!IoAllocateWorkItem
fffff880`01a35100  fffff802`b3ad7d70 nt!EtwWrite
fffff880`01a35108  fffff802`b3f6a9e0 nt!IoRegisterBootDriverReinitialization
fffff880`01a35110  fffff802`b3b06820 nt!ZwMakeTemporaryObject
fffff880`01a35118  fffff802`b3b41fd0 nt!KeReleaseMutex
fffff880`01a35120  fffff802`b3ba2140 nt!IoAllocateErrorLogEntry
fffff880`01a35128  fffff802`b3b466b0 nt!IoGetActivityIdIrp
fffff880`01a35130  fffff802`b3b8fe54 nt!IoInvalidateDeviceRelations
fffff880`01a35138  fffff802`b3e0e500 nt!EtwRegister
fffff880`01a35140  fffff802`b3b05c40 nt!ZwCreateDirectoryObject
fffff880`01a35148  fffff802`b3b3d0e0 nt!KeInitializeEvent
fffff880`01a35150  fffff802`b3f059d4 nt!MmGetSystemRoutineAddress
fffff880`01a35158  fffff802`b3ab17c0 nt!IoFreeWorkItem
fffff880`01a35160  fffff802`b3afa000 nt!KeSetEvent
fffff880`01a35168  fffff802`b3a8ddd0 nt!IoDeleteDevice
fffff880`01a35170  fffff802`b3b47190 nt!RtlInitUnicodeString
fffff880`01a35178  fffff802`b3ba8080 nt!IoSetHardErrorOrVerifyDevice
fffff880`01a35180  fffff802`b3a8d890 nt!IoReportTargetDeviceChangeAsynchronous
fffff880`01a35188  fffff802`b3e08240 nt!IoBuildSynchronousFsdRequest
fffff880`01a35190  fffff802`b3f86de0 nt!IoRegisterDriverReinitialization
fffff880`01a35198  fffff802`b3afae90 nt!strncmp
fffff880`01a351a0  fffff802`b3cf7010 nt!ExFreePoolWithTag
fffff880`01a351a8  fffff802`b3ae84e0 nt!IoBuildDeviceIoControlRequest
fffff880`01a351b0  fffff802`b3e0d890 nt!EtwUnregister
fffff880`01a351b8  fffff802`b3ba2030 nt!IoWriteErrorLogEntry
fffff880`01a351c0  fffff802`b3f994ac nt!IoWMIRegistrationControl
fffff880`01a351c8  fffff802`b3b4d300 nt!IoAllocateMdl
fffff880`01a351d0  fffff802`b3cf8040 nt!ExAllocatePoolWithTag
fffff880`01a351d8  00000000`00000000
fffff880`01a351e0  fffff880`0143e6d0 CLASSPNP!ClassInitializeSrbLookasideList
fffff880`01a351e8  fffff880`014438a4 CLASSPNP!ClassDeleteSrbLookasideList
fffff880`01a351f0  fffff880`0143f7a0 CLASSPNP!ClassInitializeMediaChangeDetection
fffff880`01a351f8  fffff880`0143eff0 CLASSPNP!ClassUpdateInformationInRegistry
fffff880`01a35200  fffff880`0143ee10 CLASSPNP!ClassGetDeviceParameter
fffff880`01a35208  fffff880`014402d0 CLASSPNP!ClassQueryTimeOutRegistryValue
fffff880`01a35210  fffff880`01401660 CLASSPNP!ClassSignalCompletion
fffff880`01a35218  fffff880`014056e0 CLASSPNP!ClassReadDriveCapacity
fffff880`01a35220  fffff880`01403540 CLASSPNP!ClassInterpretSenseInfo
fffff880`01a35228  fffff880`01408990 CLASSPNP!ClassWmiCompleteRequest
fffff880`01a35230  fffff880`0140ee70 CLASSPNP!ClassNotifyFailurePredicted
fffff880`01a35238  fffff880`014135f8 CLASSPNP!ClassReleaseQueue
fffff880`01a35240  fffff880`0143fdf0 CLASSPNP!ClassSetFailurePredictionPoll
fffff880`01a35248  fffff880`01407e10 CLASSPNP!ClassAcquireRemoveLockEx
fffff880`01a35250  fffff880`0143d440 CLASSPNP!ClassModeSense
fffff880`01a35258  fffff880`0143e5a0 CLASSPNP!ClassClaimDevice
fffff880`01a35260  fffff880`014015e0 CLASSPNP!ClassReleaseRemoveLock
fffff880`01a35268  fffff880`014091c0 CLASSPNP!ClassSpinDownPowerHandler
```

```
fffff880`01a35270  fffff880`01440180 CLASSPNP!ClassInitializeEx
fffff880`01a35278  fffff880`014049d0 CLASSPNP!ClassDeviceControl
fffff880`01a35280  fffff880`01405640 CLASSPNP!ClassCompleteRequest
fffff880`01a35288  fffff880`014042f0 CLASSPNP!ClassSendSrbSynchronous
fffff880`01a35290  fffff880`014138a0 CLASSPNP!ClassAsynchronousCompletion
fffff880`01a35298  fffff880`0144377c CLASSPNP!ClassSetDeviceParameter
fffff880`01a352a0  fffff880`0143ccc0 CLASSPNP!ClassSendDeviceIoControlSynchronous
fffff880`01a352a8  fffff880`01408b00 CLASSPNP!ClassFindModePage
fffff880`01a352b0  fffff880`01440470 CLASSPNP!ClassInitialize
fffff880`01a352b8  fffff880`01402e80 CLASSPNP!ClassIoComplete
fffff880`01a352c0  fffff880`0143e160 CLASSPNP!ClassCreateDeviceObject
fffff880`01a352c8  fffff880`0143da10 CLASSPNP!ClassScanForSpecial
fffff880`01a352d0  00000000`00000000
```

7.	We can check if there was any patching by using the **!for_each_module** command as we did for user space in the previous exercise (if you use a docker environment, please specify this command **.exepath** *C:\mss* before):

```
0: kd> !for_each_module "!chkimg -v -d @#ModuleName"
[...]
```

There are no errors.

8.	Let’s check the current thread:

```
0: kd> !thread -1 3f
THREAD fffffa8003db4740  Cid 0ca0.03e0  Teb: 000007f770b7d000 Win32Thread: fffff90104094830 RUNNING on
processor 0
Not impersonating
DeviceMap                 fffff8a007e2e6a0
Owning Process            fffffa8002d74180       Image:         Taskmgr.exe
Attached Process          N/A            Image:         N/A
Wait Start TickCount      15741128       Ticks: 0
Context Switch Count      31359          IdealProcessor: 0
UserTime                  00:00:09.859
KernelTime                00:00:07.394
Win32 Start Address taskmgr!wWinMainCRTStartup (0x000007f770e68688)
Stack Init fffff88015925dd0 Current fffff88015925800
Base fffff88015926000 Limit fffff88015920000 Call 0000000000000000
Priority 13 BasePriority 9 IoPriority 2 PagePriority 5

Child-SP          RetAddr           Call Site
fffff880`15925ae8 fffff802`b400f0dd nt!KeBugCheckEx
fffff880`15925af0 fffff802`b3ea8f6d nt!PspCatchCriticalBreak+0xad
fffff880`15925b30 fffff802`b3ea8019 nt! ?? ::NNGAKEGL::`string'+0x46f60
fffff880`15925b90 fffff802`b3ea7e52 nt!PspTerminateProcess+0x6d
fffff880`15925bd0 fffff802`b3b02d53 nt!NtTerminateProcess+0x9e
fffff880`15925c40 000007fe`f7ec2eaa nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @ fffff880`15925c40)
000000f0`6e86f3e8 000007fe`f4ff1295 ntdll!NtTerminateProcess+0xa
000000f0`6e86f3f0 000007f7`70e012ba KERNELBASE!TerminateProcess+0x25
000000f0`6e86f420 000007f7`70df3698 taskmgr!WdcProcessMonitor::OnProcessCommand+0x1b6
000000f0`6e86f4b0 000007f7`70df55bb taskmgr!WdcListView::OnProcessCommand+0x1e0
000000f0`6e86f5a0 000007f7`70df5b47 taskmgr!WdcListView::OnCommand+0x123
000000f0`6e86f5f0 000007fe`f2227239 taskmgr!WdcListView::OnMessage+0x287
000000f0`6e86f710 000007fe`f2a82d23 DUI70!DirectUI::HWNDHost::_CtrlWndProc+0xa1
000000f0`6e86f770 000007fe`f56c171e DUser!WndBridge::RawWndProc+0x73
000000f0`6e86f7e0 000007fe`f56c14d7 USER32!UserCallWinProcCheckWow+0x13a
000000f0`6e86f8a0 000007f7`70e1b0e1 USER32!DispatchMessageWorker+0x1a7
000000f0`6e86f920 000007f7`70e685e6 taskmgr!wWinMain+0x44d
000000f0`6e86fde0 000007fe`f601167e taskmgr!CBaseRPCTimeout::Disarm+0x31a
000000f0`6e86fea0 000007fe`f7ee3501 KERNEL32!BaseThreadInitThunk+0x1a
000000f0`6e86fed0 00000000`00000000 ntdll!RtlUserThreadStart+0x1d
```

Note that the small number of Ticks value may also help find threads that execute frequently or just recently spent some time executing. Also, in kernel memory dumps, we won't see user space portion of a thread stack. Here we see it because we use a complete memory dump.

9. Then we check the current process:

```
0: kd> !process -1 3f
PROCESS fffffa8002d74180
    SessionId: 2  Cid: 0ca0    Peb: 7f770b7f000  ParentCid: 0d68
    DirBase: 08818000  ObjectTable: fffff8a001f18d80  HandleCount: <Data Not Accessible>
    Image: Taskmgr.exe
    VadRoot fffffa8003e9d1e0 Vads 239 Clone 0 Private 2297. Modified 243564. Locked 0.
    DeviceMap fffff8a007e2e6a0
    Token                             fffff8a007e3b8c0
    ElapsedTime                       00:10:57.072
    UserTime                          00:00:11.325
    KernelTime                        00:00:26.878
    QuotaPoolUsage[PagedPool]         482336
    QuotaPoolUsage[NonPagedPool]      31280
    Working Set Sizes (now,min,max)  (7136, 50, 345) (28544KB, 200KB, 1380KB)
    PeakWorkingSetSize                7337
    VirtualSize                       216 Mb
    PeakVirtualSize                   343 Mb
    PageFaultCount                    51873
    MemoryPriority                    FOREGROUND
    BasePriority                      8
    CommitCharge                      2905

    PEB at 000007f770b7f000
    InheritedAddressSpace:    No
    ReadImageFileExecOptions: No
    BeingDebugged:            No
    ImageBaseAddress:         000007f770dd0000
    Ldr                       000007fef7ff88a0
    Ldr.Initialized:          Yes
    Ldr.InInitializationOrderModuleList: 000000f06e9b1a10 . 000000f070e6d150
    Ldr.InLoadOrderModuleList:           000000f06e9b1b70 . 000000f070e6d130
    Ldr.InMemoryOrderModuleList:         000000f06e9b1b80 . 000000f070e6d140
                    Base TimeStamp                     Module
             7f770dd0000 50107c26 Jul 26 00:07:18 2012 C:\WINDOWS\system32\taskmgr.exe
             7fef7ec0000 505ab405 Sep 20 07:13:25 2012 C:\WINDOWS\SYSTEM32\ntdll.dll
             7fef6010000 5010a83a Jul 26 03:15:22 2012 C:\WINDOWS\system32\KERNEL32.DLL
             7fef4fd0000 5010ab2d Jul 26 03:27:57 2012 C:\WINDOWS\system32\KERNELBASE.dll
             7fef5810000 50108b7f Jul 26 01:12:47 2012 C:\WINDOWS\system32\GDI32.dll
             7fef56c0000 505a9a92 Sep 20 05:24:50 2012 C:\WINDOWS\system32\USER32.dll
             7fef7820000 5010ac20 Jul 26 03:32:00 2012 C:\WINDOWS\system32\msvcrt.dll
             7fef5500000 50108a1d Jul 26 01:06:53 2012 C:\WINDOWS\system32\OLEAUT32.dll
             7fef52e0000 50108a89 Jul 26 01:08:41 2012 C:\WINDOWS\SYSTEM32\cfgmgr32.dll
             7fef4d90000 501089e8 Jul 26 01:06:00 2012 C:\WINDOWS\SYSTEM32\powrprof.dll
             7fef4080000 5010ac3a Jul 26 03:32:26 2012 C:\WINDOWS\system32\pcwum.dll
             7fef2760000 501084f0 Jul 26 00:44:48 2012 C:\WINDOWS\WinSxS\amd64_microsoft.windows.common-
controls_6595b64144ccf1df_6.0.9200.16384_none_418c2a697189c07f\COMCTL32.dll
             7fef3c80000 505a9614 Sep 20 05:05:40 2012 C:\WINDOWS\system32\UxTheme.dll
             7fef7ad0000 501080dd Jul 26 00:27:25 2012 C:\WINDOWS\system32\SHLWAPI.dll
             7fef6520000 507635b5 Oct 11 03:57:57 2012 C:\WINDOWS\system32\SHELL32.dll
             7fef1750000 5010969b Jul 26 02:00:11 2012 C:\WINDOWS\system32\credui.dll
             7fef2a80000 5010846e Jul 26 00:42:38 2012 C:\WINDOWS\system32\DUser.dll
             7fef21c0000 50108e6a Jul 26 01:25:14 2012 C:\WINDOWS\system32\DUI70.dll
             7feeef40000 505ab1f8 Sep 20 07:04:40 2012 C:\WINDOWS\system32\apphelp.dll
             7fef7b30000 505a9af2 Sep 20 05:26:26 2012 C:\WINDOWS\system32\combase.dll
             7fef5be0000 50108bb9 Jul 26 01:13:45 2012 C:\WINDOWS\system32\RPCRT4.dll
             7fef2ed0000 505a97e0 Sep 20 05:13:20 2012 C:\WINDOWS\system32\SHCORE.DLL
             7fef54c0000 501088ce Jul 26 01:01:18 2012 C:\WINDOWS\system32\IMM32.DLL
             7fef5d20000 50108881 Jul 26 01:00:01 2012 C:\WINDOWS\system32\MSCTF.dll
             7fef4c30000 5010ab50 Jul 26 03:28:32 2012 C:\WINDOWS\system32\CRYPTBASE.dll
             7fef4bd0000 50108a4c Jul 26 01:07:40 2012 C:\WINDOWS\system32\bcryptPrimitives.dll
             7fef2a10000 5010894e Jul 26 01:03:26 2012 C:\WINDOWS\system32\dwmapi.dll
             7fef5340000 50108270 Jul 26 00:34:08 2012 C:\WINDOWS\system32\ole32.dll
             7fef55d0000 50108a41 Jul 26 01:07:29 2012 C:\WINDOWS\SYSTEM32\sechost.dll
             7fef4d00000 5010a79e Jul 26 03:12:46 2012 C:\WINDOWS\system32\WTSAPI32.dll
             7fef4d20000 5010876c Jul 26 00:55:24 2012 C:\WINDOWS\system32\WINSTA.dll
             7feebbe0000 501089d1 Jul 26 01:05:37 2012 C:\WINDOWS\system32\srumapi.dll
             7fef5620000 501081c1 Jul 26 00:31:13 2012 C:\WINDOWS\SYSTEM32\clbcatq.dll
             7fef0b80000 505a9be8 Sep 20 05:30:32 2012 C:\WINDOWS\system32\IPHLPAPI.DLL
             7fef5330000 5010ac24 Jul 26 03:32:04 2012 C:\WINDOWS\system32\NSI.dll
             7fef0b20000 50108ad1 Jul 26 01:09:53 2012 C:\WINDOWS\system32\WINNSI.DLL
             7fef2420000 505a924c Sep 20 04:49:32 2012 C:\Windows\System32\Windows.UI.Immersive.dll
             7fef4d70000 50108a11 Jul 26 01:06:41 2012 C:\WINDOWS\system32\samcli.dll
             7fef0f50000 50108a13 Jul 26 01:06:43 2012 C:\WINDOWS\system32\SAMLIB.dll
             7fef4100000 50108a19 Jul 26 01:06:49 2012 C:\WINDOWS\system32\netutils.dll
             7fef1980000 505a9949 Sep 20 05:19:21 2012 C:\WINDOWS\system32\WindowsCodecs.dll
             7fef46a0000 50108ad9 Jul 26 01:10:01 2012 C:\WINDOWS\system32\CRYPTSP.dll
             7fef4320000 50108ac4 Jul 26 01:09:40 2012 C:\WINDOWS\system32\rsaenh.dll
             7fef26f0000 5010877b Jul 26 00:55:39 2012 C:\WINDOWS\system32\OLEACC.dll
             7fef06b0000 505a9bdc Sep 20 05:30:20 2012 C:\WINDOWS\system32\dhcpcsvc6.DLL
             7fef5b80000 50108abf Jul 26 01:09:35 2012 C:\WINDOWS\system32\WS2_32.dll
             7fef06e0000 505a9b9c Sep 20 05:29:16 2012 C:\WINDOWS\system32\dhcpcsvc.DLL
             7fef1740000 5010ac6c Jul 26 03:33:16 2012 C:\WINDOWS\system32\wlanutil.dll
             7fef03b0000 5063dc6b Sep 27 05:56:11 2012 C:\WINDOWS\system32\wlanapi.dll
             7fef37e0000 501089ec Jul 26 01:06:04 2012 C:\WINDOWS\system32\wkscli.dll
             7fef2e90000 50108843 Jul 26 00:58:59 2012 C:\WINDOWS\system32\XmlLite.dll
             7fef4df0000 50108ab9 Jul 26 01:09:29 2012 C:\WINDOWS\system32\profapi.dll
             7feed830000 501080ee Jul 26 00:27:42 2012 C:\Windows\System32\thumbcache.dll
             7fef78d0000 5010a732 Jul 26 03:10:58 2012 C:\WINDOWS\SYSTEM32\advapi32.dll
             7fef0cb0000 505a95dd Sep 20 05:04:45 2012 C:\Windows\System32\PROPSYS.dll
             7feeb9d0000 505aafdf Sep 20 06:55:43 2012 C:\Windows\System32\actxprxy.dll
             7fef2580000 501089b7 Jul 26 01:05:11 2012 C:\WINDOWS\system32\Bcp47Langs.dll
             7fef48f0000 50108aca Jul 26 01:09:46 2012 C:\WINDOWS\SYSTEM32\bcrypt.dll
             7feeeb70000 50107f98 Jul 26 00:22:00 2012 C:\Windows\System32\MrmCoreR.dll
             7fef7d60000 505a9257 Sep 20 04:49:43 2012 C:\WINDOWS\system32\urlmon.dll
             7fef6160000 505aa96c Sep 20 06:28:12 2012 C:\WINDOWS\system32\iertutil.dll
             7fef5950000 505a9365 Sep 20 04:54:13 2012 C:\WINDOWS\system32\WININET.dll
             7fef5e40000 501080fc Jul 26 00:27:56 2012 C:\WINDOWS\system32\SETUPAPI.dll
             7fef50d0000 5010898b Jul 26 01:04:27 2012 C:\WINDOWS\system32\DEVOBJ.dll
             7fee8a40000 505a9555 Sep 20 05:02:29 2012 C:\Windows\System32\twinapi.dll
             7fef31b0000 50108834 Jul 26 00:58:44 2012 C:\WINDOWS\system32\dbghelp.dll
             7feeb770000 50109564 Jul 26 01:55:00 2012 C:\WINDOWS\System32\cscui.dll
             7fef30c0000 5010a9be Jul 26 03:21:50 2012 C:\WINDOWS\System32\CSCDLL.dll
             7fef30d0000 5010a183 Jul 26 02:46:43 2012 C:\WINDOWS\System32\cscobj.dll
             7fef4420000 50108843 Jul 26 00:58:59 2012 C:\WINDOWS\System32\USERENV.dll
             7feec150000 501089ad Jul 26 01:05:01 2012 C:\WINDOWS\system32\CSCAPI.dll
             7fee72f0000 50109745 Jul 26 02:03:01 2012 C:\Windows\System32\EhStorShell.dll
             7feef920000 501089fe Jul 26 01:06:22 2012 C:\WINDOWS\SYSTEM32\ntmarta.dll
             7feeb240000 501081d7 Jul 26 00:31:35 2012 C:\WINDOWS\SYSTEM32\profext.dll
             7fef4ba0000 505a9be9 Sep 20 05:30:33 2012 C:\WINDOWS\system32\SSPICLI.DLL
             7fef3320000 50108655 Jul 26 00:50:45 2012 C:\Windows\System32\taskschd.dll
    SubSystemData:     0000000000000000
    ProcessHeap:       000000f06e9b0000
    ProcessParameters: 000000f06e9b11e0
    CurrentDirectory:  'C:\WINDOWS\system32\'
    WindowTitle:  'C:\WINDOWS\system32\taskmgr.exe'
    ImageFile:    'C:\WINDOWS\system32\taskmgr.exe'
    CommandLine:  '"C:\WINDOWS\system32\taskmgr.exe" /4'
    DllPath:      '< Name not readable >'
    Environment:  000000f06e9b0860
        ALLUSERSPROFILE=C:\ProgramData
        APPDATA=C:\Users\Dmitry\AppData\Roaming
        CommonProgramFiles=C:\Program Files\Common Files
        CommonProgramFiles(x86)=C:\Program Files (x86)\Common Files
        CommonProgramW6432=C:\Program Files\Common Files
        COMPUTERNAME=MACAIR1
        ComSpec=C:\WINDOWS\system32\cmd.exe
        FP_NO_HOST_CHECK=NO
        HOMEDRIVE=C:
        HOMEPATH=\Users\Dmitry
        LOCALAPPDATA=C:\Users\Dmitry\AppData\Local
        LOGONSERVER=\\MicrosoftAccount
        NUMBER_OF_PROCESSORS=2
        OS=Windows_NT
        Path=C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WINDOWS\System32\WindowsPowerShell\v1.0\
        PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
        PROCESSOR_ARCHITECTURE=AMD64
        PROCESSOR_IDENTIFIER=Intel64 Family 6 Model 15 Stepping 11, GenuineIntel
        PROCESSOR_LEVEL=6
        PROCESSOR_REVISION=0f0b
        ProgramData=C:\ProgramData
        ProgramFiles=C:\Program Files
        ProgramFiles(x86)=C:\Program Files (x86)
        ProgramW6432=C:\Program Files
        PSModulePath=C:\WINDOWS\system32\WindowsPowerShell\v1.0\Modules\
        PUBLIC=C:\Users\Public
        SystemDrive=C:
        SystemRoot=C:\WINDOWS
        TEMP=C:\Users\Dmitry\AppData\Local\Temp
        TMP=C:\Users\Dmitry\AppData\Local\Temp
        USERDOMAIN=MACAIR1
        USERDOMAIN_ROAMINGPROFILE=MACAIR1
        USERNAME=Dmitry
        USERPROFILE=C:\Users\Dmitry
        windir=C:\WINDOWS


        THREAD fffffa8003db4740  Cid 0ca0.03e0  Teb: 000007f770b7d000 Win32Thread: fffff90104094830 RUNNING on
processor 0
        Not impersonating
        DeviceMap                 fffff8a007e2e6a0
        Owning Process            fffffa8002d74180       Image:         Taskmgr.exe
        Attached Process          N/A            Image:         N/A
        Wait Start TickCount      15741128       Ticks: 0
        Context Switch Count      31359          IdealProcessor: 0
        UserTime                  00:00:09.859
        KernelTime                00:00:07.394
        Win32 Start Address taskmgr!wWinMainCRTStartup (0x000007f770e68688)
        Stack Init fffff88015925dd0 Current fffff88015925800
        Base fffff88015926000 Limit fffff88015920000 Call 0000000000000000
        Priority 13 BasePriority 9 PriorityDecrement 2 IoPriority 2 PagePriority 5

        Child-SP          RetAddr           Call Site
        fffff880`15925ae8 fffff802`b400f0dd nt!KeBugCheckEx
        fffff880`15925af0 fffff802`b3ea8f6d nt!PspCatchCriticalBreak+0xad
        fffff880`15925b30 fffff802`b3ea8019 nt! ?? ::NNGAKEGL::`string'+0x46f60
        fffff880`15925b90 fffff802`b3ea7e52 nt!PspTerminateProcess+0x6d
        fffff880`15925bd0 fffff802`b3b02d53 nt!NtTerminateProcess+0x9e
        fffff880`15925c40 000007fe`f7ec2eaa nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @ fffff880`15925c40)
        000000f0`6e86f3e8 000007fe`f4ff1295 ntdll!NtTerminateProcess+0xa
        000000f0`6e86f3f0 000007f7`70e012ba KERNELBASE!TerminateProcess+0x25
        000000f0`6e86f420 000007f7`70df3698 taskmgr!WdcProcessMonitor::OnProcessCommand+0x1b6
        000000f0`6e86f4b0 000007f7`70df55bb taskmgr!WdcListView::OnProcessCommand+0x1e0
        000000f0`6e86f5a0 000007f7`70df5b47 taskmgr!WdcListView::OnCommand+0x123
        000000f0`6e86f5f0 000007fe`f2227239 taskmgr!WdcListView::OnMessage+0x287
        000000f0`6e86f710 000007fe`f2a82d23 DUI70!DirectUI::HWNDHost::_CtrlWndProc+0xa1
        000000f0`6e86f770 000007fe`f56c171e DUser!WndBridge::RawWndProc+0x73
        000000f0`6e86f7e0 000007fe`f56c14d7 USER32!UserCallWinProcCheckWow+0x13a
        000000f0`6e86f8a0 000007f7`70e1b0e1 USER32!DispatchMessageWorker+0x1a7
        000000f0`6e86f920 000007f7`70e685e6 taskmgr!wWinMain+0x44d
        000000f0`6e86fde0 000007fe`f601167e taskmgr!CBaseRPCTimeout::Disarm+0x31a
        000000f0`6e86fea0 000007fe`f7ee3501 KERNEL32!BaseThreadInitThunk+0x1a
        000000f0`6e86fed0 00000000`00000000 ntdll!RtlUserThreadStart+0x1d

        THREAD fffffa80039dfb00  Cid 0ca0.0564  Teb: 000007f770b7b000 Win32Thread: fffff90103f44710 WAIT:
(UserRequest) UserMode Non-Alertable
            fffffa8003665fe0  SynchronizationEvent
            fffffa8002cc1d30  SynchronizationEvent
        Not impersonating
        DeviceMap                 fffff8a007e2e6a0
        Owning Process            fffffa8002d74180       Image:         Taskmgr.exe
        Attached Process          N/A            Image:         N/A
        Wait Start TickCount      15699020       Ticks: 42108 (0:00:10:56.889)
        Context Switch Count      4              IdealProcessor: 0
        UserTime                  00:00:00.000
        KernelTime                00:00:00.000
        Win32 Start Address msvcrt!endthreadex (0x000007fef7845e10)
        Stack Init fffff880155d5dd0 Current fffff880155d5180
        Base fffff880155d6000 Limit fffff880155d0000 Call 0000000000000000
        Priority 9 BasePriority 8 PriorityDecrement 0 IoPriority 2 PagePriority 5
        Kernel stack not resident.
        Child-SP          RetAddr           Call Site
        fffff880`155d51c0 fffff802`b3b2d99c nt!KiSwapContext+0x76
        fffff880`155d5300 fffff802`b3b293cd nt!KiCommitThreadWait+0x23c
        fffff880`155d53c0 fffff802`b3eca2ac nt!KeWaitForMultipleObjects+0x25d
        fffff880`155d5470 fffff802`b3eca723 nt!ObWaitForMultipleObjects+0x29c
        fffff880`155d5980 fffff802`b3b02d53 nt!NtWaitForMultipleObjects+0xe3
        fffff880`155d5bd0 000007fe`f7ec319b nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @ fffff880`155d5c40)
        000000f0`7025f938 000007fe`f4fd12c6 ntdll!NtWaitForMultipleObjects+0xa
        000000f0`7025f940 000007fe`f56c2c83 KERNELBASE!WaitForMultipleObjectsEx+0xe5
        000000f0`7025fc20 000007fe`f2aa160b USER32!MsgWaitForMultipleObjectsEx+0x144
        000000f0`7025fcd0 000007fe`f2aa15db DUser!CoreSC::xwProcessNL+0x5bb
        000000f0`7025fda0 000007fe`f2aa14fe DUser!GetMessageExA+0x6b
        000000f0`7025fdf0 000007fe`f782707b DUser!ResourceManager::SharedThreadProc+0xfe
        000000f0`7025fe80 000007fe`f7845e6d msvcrt!endthreadex+0xcb
        000000f0`7025feb0 000007fe`f601167e msvcrt!endthreadex+0xac
        000000f0`7025fee0 000007fe`f7ee3501 KERNEL32!BaseThreadInitThunk+0x1a
        000000f0`7025ff10 00000000`00000000 ntdll!RtlUserThreadStart+0x1d

        THREAD fffffa8003253b00  Cid 0ca0.0d64  Teb: 000007f770b79000 Win32Thread: 0000000000000000 WAIT:
(UserRequest) UserMode Non-Alertable
            fffffa800307aca0  NotificationEvent
            fffffa80036357a0  SynchronizationEvent
        Not impersonating
        DeviceMap                 fffff8a007e2e6a0
        Owning Process            fffffa8002d74180       Image:         Taskmgr.exe
        Attached Process          N/A            Image:         N/A
        Wait Start TickCount      15741108       Ticks: 20 (0:00:00:00.312)
        Context Switch Count      653            IdealProcessor: 1
        UserTime                  00:00:00.000
        KernelTime                00:00:00.000
        Win32 Start Address taskmgr!WdcDataMonitor::UpdateThread (0x000007f770dfdf1c)
        Stack Init fffff880159dadd0 Current fffff880159da180
        Base fffff880159db000 Limit fffff880159d5000 Call 0000000000000000
        Priority 11 BasePriority 8 PriorityDecrement 2 IoPriority 2 PagePriority 5
        Child-SP          RetAddr           Call Site
        fffff880`159da1c0 fffff802`b3b2d99c nt!KiSwapContext+0x76
        fffff880`159da300 fffff802`b3b293cd nt!KiCommitThreadWait+0x23c
        fffff880`159da3c0 fffff802`b3eca2ac nt!KeWaitForMultipleObjects+0x25d
        fffff880`159da470 fffff802`b3eca723 nt!ObWaitForMultipleObjects+0x29c
        fffff880`159da980 fffff802`b3b02d53 nt!NtWaitForMultipleObjects+0xe3
        fffff880`159dabd0 000007fe`f7ec319b nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @ fffff880`159dac40)
        000000f0`7238f4f8 000007fe`f4fd12c6 ntdll!NtWaitForMultipleObjects+0xa
        000000f0`7238f500 000007fe`f6011292 KERNELBASE!WaitForMultipleObjectsEx+0xe5
        000000f0`7238f7e0 000007f7`70dfdc81 KERNEL32!WaitForMultipleObjects+0x12
        000000f0`7238f820 000007f7`70dfdf54 taskmgr!WdcDataMonitor::DoUpdates+0x3d
        000000f0`7238f860 000007fe`f601167e taskmgr!WdcDataMonitor::UpdateThread+0x38
        000000f0`7238f8a0 000007fe`f7ee3501 KERNEL32!BaseThreadInitThunk+0x1a
        000000f0`7238f8d0 00000000`00000000 ntdll!RtlUserThreadStart+0x1d

        THREAD fffffa8003b45b00  Cid 0ca0.0824  Teb: 000007f770b77000 Win32Thread: fffff90103f5cb90 WAIT:
(UserRequest) UserMode Non-Alertable
            fffffa8003612250  NotificationEvent
            fffffa8002cb6890  SynchronizationEvent
        Not impersonating
        DeviceMap                 fffff8a007e2e6a0
        Owning Process            fffffa8002d74180       Image:         Taskmgr.exe
        Attached Process          N/A            Image:         N/A
        Wait Start TickCount      15741108       Ticks: 20 (0:00:00:00.312)
        Context Switch Count      2818           IdealProcessor: 0
        UserTime                  00:00:00.031
        KernelTime                00:00:00.124
        Win32 Start Address taskmgr!WdcDataMonitor::UpdateThread (0x000007f770dfdf1c)
        Stack Init fffff8801595ddd0 Current fffff8801595d180
        Base fffff8801595e000 Limit fffff88015958000 Call 0000000000000000
        Priority 13 BasePriority 10 PriorityDecrement 2 IoPriority 2 PagePriority 5
        Child-SP          RetAddr           Call Site
        fffff880`1595d1c0 fffff802`b3b2d99c nt!KiSwapContext+0x76
        fffff880`1595d300 fffff802`b3b293cd nt!KiCommitThreadWait+0x23c
        fffff880`1595d3c0 fffff802`b3eca2ac nt!KeWaitForMultipleObjects+0x25d
        fffff880`1595d470 fffff802`b3eca723 nt!ObWaitForMultipleObjects+0x29c
        fffff880`1595d980 fffff802`b3b02d53 nt!NtWaitForMultipleObjects+0xe3
        fffff880`1595dbd0 000007fe`f7ec319b nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @ fffff880`1595dc40)
        000000f0`7240f9f8 000007fe`f4fd12c6 ntdll!NtWaitForMultipleObjects+0xa
        000000f0`7240fa00 000007fe`f6011292 KERNELBASE!WaitForMultipleObjectsEx+0xe5
        000000f0`7240fce0 000007f7`70dfdc81 KERNEL32!WaitForMultipleObjects+0x12
        000000f0`7240fd20 000007f7`70dfdf54 taskmgr!WdcDataMonitor::DoUpdates+0x3d
        000000f0`7240fd60 000007fe`f601167e taskmgr!WdcDataMonitor::UpdateThread+0x38
        000000f0`7240fda0 000007fe`f7ee3501 KERNEL32!BaseThreadInitThunk+0x1a
        000000f0`7240fdd0 00000000`00000000 ntdll!RtlUserThreadStart+0x1d

        THREAD fffffa80018eab00  Cid 0ca0.0888  Teb: 000007f770b75000 Win32Thread: fffff90103ff8b90 WAIT:
(UserRequest) UserMode Non-Alertable
            fffffa8001c81ca0  NotificationEvent
            fffffa80036767a0  SynchronizationEvent
        Not impersonating
        DeviceMap                 fffff8a007e2e6a0
        Owning Process            fffffa8002d74180       Image:         Taskmgr.exe
        Attached Process          N/A            Image:         N/A
        Wait Start TickCount      15741108       Ticks: 20 (0:00:00:00.312)
        Context Switch Count      4747           IdealProcessor: 1
        UserTime                  00:00:00.000
        KernelTime                00:00:00.078
        Win32 Start Address taskmgr!WdcDataMonitor::UpdateThread (0x000007f770dfdf1c)
        Stack Init fffff8801594fdd0 Current fffff8801594f180
        Base fffff88015950000 Limit fffff8801594a000 Call 0000000000000000
        Priority 11 BasePriority 8 PriorityDecrement 2 IoPriority 2 PagePriority 5
        Child-SP          RetAddr           Call Site
        fffff880`1594f1c0 fffff802`b3b2d99c nt!KiSwapContext+0x76
        fffff880`1594f300 fffff802`b3b293cd nt!KiCommitThreadWait+0x23c
        fffff880`1594f3c0 fffff802`b3eca2ac nt!KeWaitForMultipleObjects+0x25d
        fffff880`1594f470 fffff802`b3eca723 nt!ObWaitForMultipleObjects+0x29c
        fffff880`1594f980 fffff802`b3b02d53 nt!NtWaitForMultipleObjects+0xe3
        fffff880`1594fbd0 000007fe`f7ec319b nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @ fffff880`1594fc40)
        000000f0`7248f548 000007fe`f4fd12c6 ntdll!NtWaitForMultipleObjects+0xa
        000000f0`7248f550 000007fe`f6011292 KERNELBASE!WaitForMultipleObjectsEx+0xe5
        000000f0`7248f830 000007f7`70dfdc81 KERNEL32!WaitForMultipleObjects+0x12
        000000f0`7248f870 000007f7`70dfdf54 taskmgr!WdcDataMonitor::DoUpdates+0x3d
        000000f0`7248f8b0 000007fe`f601167e taskmgr!WdcDataMonitor::UpdateThread+0x38
        000000f0`7248f8f0 000007fe`f7ee3501 KERNEL32!BaseThreadInitThunk+0x1a
        000000f0`7248f920 00000000`00000000 ntdll!RtlUserThreadStart+0x1d

        THREAD fffffa80033f63c0  Cid 0ca0.0e28  Teb: 000007f770b73000 Win32Thread: fffff901006bb710 WAIT:
(UserRequest) UserMode Non-Alertable
            fffffa80040844b0  NotificationEvent
            fffffa8002e58710  SynchronizationEvent
        Not impersonating
        DeviceMap                 fffff8a007e2e6a0
        Owning Process            fffffa8002d74180       Image:         Taskmgr.exe
        Attached Process          N/A            Image:         N/A
        Wait Start TickCount      15699023       Ticks: 42105 (0:00:10:56.842)
        Context Switch Count      6              IdealProcessor: 0
        UserTime                  00:00:00.000
        KernelTime                00:00:00.000
        Win32 Start Address taskmgr!WdcDataMonitor::UpdateThread (0x000007f770dfdf1c)
        Stack Init fffff880159ccdd0 Current fffff880159cc180
        Base fffff880159cd000 Limit fffff880159c7000 Call 0000000000000000
        Priority 11 BasePriority 8 PriorityDecrement 2 IoPriority 2 PagePriority 5
        Kernel stack not resident.
        Child-SP          RetAddr           Call Site
        fffff880`159cc1c0 fffff802`b3b2d99c nt!KiSwapContext+0x76
        fffff880`159cc300 fffff802`b3b293cd nt!KiCommitThreadWait+0x23c
        fffff880`159cc3c0 fffff802`b3eca2ac nt!KeWaitForMultipleObjects+0x25d
        fffff880`159cc470 fffff802`b3eca723 nt!ObWaitForMultipleObjects+0x29c
        fffff880`159cc980 fffff802`b3b02d53 nt!NtWaitForMultipleObjects+0xe3
        fffff880`159ccbd0 000007fe`f7ec319b nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @ fffff880`159ccc40)
        000000f0`7250f448 000007fe`f4fd12c6 ntdll!NtWaitForMultipleObjects+0xa
        000000f0`7250f450 000007fe`f56c2c83 KERNELBASE!WaitForMultipleObjectsEx+0xe5
        000000f0`7250f730 000007f7`70e43c03 USER32!MsgWaitForMultipleObjectsEx+0x144
        000000f0`7250f7e0 000007f7`70dfdf54 taskmgr!WdcAppHistoryMonitor::DoUpdates+0x3f
        000000f0`7250f850 000007fe`f601167e taskmgr!WdcDataMonitor::UpdateThread+0x38
        000000f0`7250f890 000007fe`f7ee3501 KERNEL32!BaseThreadInitThunk+0x1a
        000000f0`7250f8c0 00000000`00000000 ntdll!RtlUserThreadStart+0x1d

        THREAD fffffa8001f075c0  Cid 0ca0.06d4  Teb: 000007f770a4c000 Win32Thread: fffff901040b5b90 WAIT:
(UserRequest) UserMode Non-Alertable
            fffffa8002d94de0  NotificationEvent
            fffffa800371fc70  SynchronizationEvent
            fffffa8002d704f0  SynchronizationEvent
        Not impersonating
        DeviceMap                 fffff8a007e2e6a0
        Owning Process            fffffa8002d74180       Image:         Taskmgr.exe
        Attached Process          N/A            Image:         N/A
        Wait Start TickCount      15741108       Ticks: 20 (0:00:00:00.312)
        Context Switch Count      19727          IdealProcessor: 1
        UserTime                  00:00:00.000
        KernelTime                00:00:00.078
        Win32 Start Address taskmgr!TmTraceControl::IncrementThread (0x000007f770df1fc4)
        Stack Init fffff880159efdd0 Current fffff880159ef180
        Base fffff880159f0000 Limit fffff880159ea000 Call 0000000000000000
        Priority 11 BasePriority 8 PriorityDecrement 2 IoPriority 2 PagePriority 5
        Child-SP          RetAddr           Call Site
        fffff880`159ef1c0 fffff802`b3b2d99c nt!KiSwapContext+0x76
        fffff880`159ef300 fffff802`b3b293cd nt!KiCommitThreadWait+0x23c
        fffff880`159ef3c0 fffff802`b3eca2ac nt!KeWaitForMultipleObjects+0x25d
        fffff880`159ef470 fffff802`b3eca723 nt!ObWaitForMultipleObjects+0x29c
        fffff880`159ef980 fffff802`b3b02d53 nt!NtWaitForMultipleObjects+0xe3
        fffff880`159efbd0 000007fe`f7ec319b nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @ fffff880`159efc40)
        000000f0`7260fb58 000007fe`f4fd12c6 ntdll!NtWaitForMultipleObjects+0xa
        000000f0`7260fb60 000007fe`f6011292 KERNELBASE!WaitForMultipleObjectsEx+0xe5
        000000f0`7260fe40 000007f7`70df2118 KERNEL32!WaitForMultipleObjects+0x12
        000000f0`7260fe80 000007fe`f601167e taskmgr!TmTraceControl::IncrementThreadInternal+0x148
        000000f0`7260ff30 000007fe`f7ee3501 KERNEL32!BaseThreadInitThunk+0x1a
        000000f0`7260ff60 00000000`00000000 ntdll!RtlUserThreadStart+0x1d

        THREAD fffffa8003f23b00  Cid 0ca0.0db8  Teb: 000007f770a4a000 Win32Thread: fffff90103fa5610 WAIT:
(UserRequest) UserMode Non-Alertable
            fffffa80036d1420  NotificationEvent
            fffffa80036c8cb0  SynchronizationEvent
        Not impersonating
        DeviceMap                 fffff8a007e2e6a0
        Owning Process            fffffa8002d74180       Image:         Taskmgr.exe
        Attached Process          N/A            Image:         N/A
        Wait Start TickCount      15741106       Ticks: 22 (0:00:00:00.343)
        Context Switch Count      811            IdealProcessor: 1
        UserTime                  00:00:00.000
        KernelTime                00:00:00.000
        Win32 Start Address taskmgr!CRUMAPIHelper::SrumThread (0x000007f770e0db10)
        Stack Init fffff88015e0ddd0 Current fffff88015e0d180
        Base fffff88015e0e000 Limit fffff88015e08000 Call 0000000000000000
        Priority 11 BasePriority 8 PriorityDecrement 2 IoPriority 2 PagePriority 5
        Child-SP          RetAddr           Call Site
        fffff880`15e0d1c0 fffff802`b3b2d99c nt!KiSwapContext+0x76
        fffff880`15e0d300 fffff802`b3b293cd nt!KiCommitThreadWait+0x23c
        fffff880`15e0d3c0 fffff802`b3eca2ac nt!KeWaitForMultipleObjects+0x25d
        fffff880`15e0d470 fffff802`b3eca723 nt!ObWaitForMultipleObjects+0x29c
        fffff880`15e0d980 fffff802`b3b02d53 nt!NtWaitForMultipleObjects+0xe3
        fffff880`15e0dbd0 000007fe`f7ec319b nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @ fffff880`15e0dc40)
        000000f0`7268f4b8 000007fe`f4fd12c6 ntdll!NtWaitForMultipleObjects+0xa
        000000f0`7268f4c0 000007fe`f56c2c83 KERNELBASE!WaitForMultipleObjectsEx+0xe5
        000000f0`7268f7a0 000007f7`70e0dd3a USER32!MsgWaitForMultipleObjectsEx+0x144
        000000f0`7268f850 000007fe`f601167e taskmgr!CRUMAPIHelper::SrumThread+0x22a
        000000f0`7268f940 000007fe`f7ee3501 KERNEL32!BaseThreadInitThunk+0x1a
        000000f0`7268f970 00000000`00000000 ntdll!RtlUserThreadStart+0x1d

        THREAD fffffa800404a080  Cid 0ca0.0c88  Teb: 000007f770a48000 Win32Thread: fffff901006b9710 WAIT:
(UserRequest) UserMode Non-Alertable
            fffffa8001c95500  NotificationEvent
            fffffa8003f37990  SynchronizationEvent
            fffffa800409e6c0  SynchronizationEvent
        Not impersonating
        DeviceMap                 fffff8a007e2e6a0
        Owning Process            fffffa8002d74180       Image:         Taskmgr.exe
        Attached Process          N/A            Image:         N/A
        Wait Start TickCount      15699025       Ticks: 42103 (0:00:10:56.811)
        Context Switch Count      7              IdealProcessor: 0
        UserTime                  00:00:00.000
        KernelTime                00:00:00.000
        Win32 Start Address taskmgr!WdcDataMonitor::UpdateThread (0x000007f770dfdf1c)
        Stack Init fffff88015e22dd0 Current fffff88015e22180
        Base fffff88015e23000 Limit fffff88015e1d000 Call 0000000000000000
        Priority 11 BasePriority 8 PriorityDecrement 2 IoPriority 2 PagePriority 5
        Kernel stack not resident.
        Child-SP          RetAddr           Call Site
        fffff880`15e221c0 fffff802`b3b2d99c nt!KiSwapContext+0x76
        fffff880`15e22300 fffff802`b3b293cd nt!KiCommitThreadWait+0x23c
        fffff880`15e223c0 fffff802`b3eca2ac nt!KeWaitForMultipleObjects+0x25d
        fffff880`15e22470 fffff802`b3eca723 nt!ObWaitForMultipleObjects+0x29c
        fffff880`15e22980 fffff802`b3b02d53 nt!NtWaitForMultipleObjects+0xe3
        fffff880`15e22bd0 000007fe`f7ec319b nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @ fffff880`15e22c40)
        000000f0`7270f448 000007fe`f4fd12c6 ntdll!NtWaitForMultipleObjects+0xa
        000000f0`7270f450 000007fe`f56c2c83 KERNELBASE!WaitForMultipleObjectsEx+0xe5
        000000f0`7270f730 000007f7`70e475fd USER32!MsgWaitForMultipleObjectsEx+0x144
        000000f0`7270f7e0 000007f7`70dfdf54 taskmgr!WdcUserMonitor::DoUpdates+0x65
        000000f0`7270f870 000007fe`f601167e taskmgr!WdcDataMonitor::UpdateThread+0x38
        000000f0`7270f8b0 000007fe`f7ee3501 KERNEL32!BaseThreadInitThunk+0x1a
        000000f0`7270f8e0 00000000`00000000 ntdll!RtlUserThreadStart+0x1d
        THREAD fffffa8001de0b00  Cid 0ca0.0c84  Teb: 000007f770a46000 Win32Thread: fffff9010065f010 WAIT:
(UserRequest) UserMode Non-Alertable
            fffffa800372dc50  NotificationEvent
            fffffa80041961c0  SynchronizationEvent
        Not impersonating
        DeviceMap                 fffff8a007e2e6a0
        Owning Process            fffffa8002d74180       Image:         Taskmgr.exe
        Attached Process          N/A            Image:         N/A
        Wait Start TickCount      15741108       Ticks: 20 (0:00:00:00.312)
        Context Switch Count      2887           IdealProcessor: 1
        UserTime                  00:00:00.015
        KernelTime                00:00:00.000
        Win32 Start Address taskmgr!WdcDataMonitor::UpdateThread (0x000007f770dfdf1c)
        Stack Init fffff88015e29dd0 Current fffff88015e29180
        Base fffff88015e2a000 Limit fffff88015e24000 Call 0000000000000000
        Priority 11 BasePriority 8 PriorityDecrement 2 IoPriority 2 PagePriority 5
        Child-SP          RetAddr           Call Site
        fffff880`15e291c0 fffff802`b3b2d99c nt!KiSwapContext+0x76
        fffff880`15e29300 fffff802`b3b293cd nt!KiCommitThreadWait+0x23c
        fffff880`15e293c0 fffff802`b3eca2ac nt!KeWaitForMultipleObjects+0x25d
        fffff880`15e29470 fffff802`b3eca723 nt!ObWaitForMultipleObjects+0x29c
        fffff880`15e29980 fffff802`b3b02d53 nt!NtWaitForMultipleObjects+0xe3
        fffff880`15e29bd0 000007fe`f7ec319b nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @ fffff880`15e29c40)
        000000f0`7278f348 000007fe`f4fd12c6 ntdll!NtWaitForMultipleObjects+0xa
        000000f0`7278f350 000007fe`f56c2c83 KERNELBASE!WaitForMultipleObjectsEx+0xe5
        000000f0`7278f630 000007f7`70e43c03 USER32!MsgWaitForMultipleObjectsEx+0x144
        000000f0`7278f6e0 000007f7`70dfdf54 taskmgr!WdcAppHistoryMonitor::DoUpdates+0x3f
        000000f0`7278f750 000007fe`f601167e taskmgr!WdcDataMonitor::UpdateThread+0x38
        000000f0`7278f790 000007fe`f7ee3501 KERNEL32!BaseThreadInitThunk+0x1a
        000000f0`7278f7c0 00000000`00000000 ntdll!RtlUserThreadStart+0x1d

        THREAD fffffa80039d3b00  Cid 0ca0.07e4  Teb: 000007f770a44000 Win32Thread: fffff901040e2530 WAIT:
(UserRequest) UserMode Non-Alertable
            fffffa8002067370  SynchronizationEvent
            fffffa8003f46e10  NotificationEvent
            fffffa800205cce0  SynchronizationEvent
            fffffa8003826490  SynchronizationEvent
            fffffa8003ee0dc0  SynchronizationEvent
            fffffa80030959b8  NotificationEvent
            fffffa800362fd18  NotificationEvent
        IRP List:
            fffffa800211ac10: (0006,03e8) Flags: 00060000  Mdl: 00000000
            fffffa800198a360: (0006,03e8) Flags: 00060000  Mdl: 00000000
        Not impersonating
        DeviceMap                 fffff8a007e2e6a0
        Owning Process            fffffa8002d74180       Image:         Taskmgr.exe
        Attached Process          N/A            Image:         N/A
        Wait Start TickCount      15699048       Ticks: 42080 (0:00:10:56.452)
        Context Switch Count      40             IdealProcessor: 0
        UserTime                  00:00:00.000
        KernelTime                00:00:00.000
        Win32 Start Address taskmgr!WdcDataMonitor::UpdateThread (0x000007f770dfdf1c)
        Stack Init fffff88015e3edd0 Current fffff88015e3e180
        Base fffff88015e3f000 Limit fffff88015e39000 Call 0000000000000000
        Priority 11 BasePriority 8 PriorityDecrement 2 IoPriority 2 PagePriority 5
        Kernel stack not resident.
        Child-SP          RetAddr           Call Site
        fffff880`15e3e1c0 fffff802`b3b2d99c nt!KiSwapContext+0x76
        fffff880`15e3e300 fffff802`b3b293cd nt!KiCommitThreadWait+0x23c
        fffff880`15e3e3c0 fffff802`b3eca2ac nt!KeWaitForMultipleObjects+0x25d
        fffff880`15e3e470 fffff802`b3eca723 nt!ObWaitForMultipleObjects+0x29c
        fffff880`15e3e980 fffff802`b3b02d53 nt!NtWaitForMultipleObjects+0xe3
        fffff880`15e3ebd0 000007fe`f7ec319b nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @ fffff880`15e3ec40)
        000000f0`7280f588 000007fe`f4fd12c6 ntdll!NtWaitForMultipleObjects+0xa
        000000f0`7280f590 000007fe`f6011292 KERNELBASE!WaitForMultipleObjectsEx+0xe5
        000000f0`7280f870 000007f7`70e57ed5 KERNEL32!WaitForMultipleObjects+0x12
        000000f0`7280f8b0 000007f7`70dfdf54 taskmgr!WdcStartupMonitor::DoUpdates+0x2ad
        000000f0`7280fdc0 000007fe`f601167e taskmgr!WdcDataMonitor::UpdateThread+0x38
        000000f0`7280fe00 000007fe`f7ee3501 KERNEL32!BaseThreadInitThunk+0x1a
        000000f0`7280fe30 00000000`00000000 ntdll!RtlUserThreadStart+0x1d

        THREAD fffffa8002d01200  Cid 0ca0.0a9c  Teb: 000007f770a42000 Win32Thread: fffff901040f7b90 WAIT: (WrQueue)
UserMode Alertable
            fffffa8001e75ec0  QueueObject
        Not impersonating
        DeviceMap                 fffff8a007e2e6a0
        Owning Process            fffffa8002d74180       Image:         Taskmgr.exe
        Attached Process          N/A            Image:         N/A
        Wait Start TickCount      15740913       Ticks: 215 (0:00:00:03.354)
        Context Switch Count      565            IdealProcessor: 0
        UserTime                  00:00:00.000
        KernelTime                00:00:00.000
        Win32 Start Address ntdll!TppWorkerThread (0x000007fef7ee38c0)
        Stack Init fffff88015e4cdd0 Current fffff88015e4c760
        Base fffff88015e4d000 Limit fffff88015e47000 Call 0000000000000000
        Priority 10 BasePriority 8 PriorityDecrement 2 IoPriority 2 PagePriority 5
        Child-SP          RetAddr           Call Site
        fffff880`15e4c7a0 fffff802`b3b2d99c nt!KiSwapContext+0x76
        fffff880`15e4c8e0 fffff802`b3b38ddb nt!KiCommitThreadWait+0x23c
        fffff880`15e4c9a0 fffff802`b3ed0b6c nt!KeRemoveQueueEx+0x26b
        fffff880`15e4ca50 fffff802`b3b434d5 nt!IoRemoveIoCompletion+0x4c
        fffff880`15e4cae0 fffff802`b3b02d53 nt!NtWaitForWorkViaWorkerFactory+0x295
        fffff880`15e4cc40 000007fe`f7ec46ab nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @ fffff880`15e4cc40)
        000000f0`7288f808 000007fe`f7ec84b3 ntdll!ZwWaitForWorkViaWorkerFactory+0xa
        000000f0`7288f810 000007fe`f601167e ntdll!TppWorkerThread+0x275
        000000f0`7288fab0 000007fe`f7ee3501 KERNEL32!BaseThreadInitThunk+0x1a
        000000f0`7288fae0 00000000`00000000 ntdll!RtlUserThreadStart+0x1d

        THREAD fffffa80040036c0  Cid 0ca0.0244  Teb: 000007f770a3c000 Win32Thread: 0000000000000000 WAIT:
(UserRequest) UserMode Non-Alertable
            fffffa80021566a0  SynchronizationEvent
            fffffa8002cd3ce0  SynchronizationEvent
        Not impersonating
        DeviceMap                 fffff8a007e2e6a0
        Owning Process            fffffa8002d74180       Image:         Taskmgr.exe
        Attached Process          N/A            Image:         N/A
        Wait Start TickCount      15739266       Ticks: 1862 (0:00:00:29.047)
        Context Switch Count      1896           IdealProcessor: 1
        UserTime                  00:00:00.015
        KernelTime                00:00:00.000
        Win32 Start Address taskmgr!WdcServiceCache::s_InformClientsThread (0x000007f770e07be4)
        Stack Init fffff88015f10dd0 Current fffff88015f10180
        Base fffff88015f11000 Limit fffff88015f0b000 Call 0000000000000000
        Priority 11 BasePriority 8 PriorityDecrement 2 IoPriority 2 PagePriority 5
        Child-SP          RetAddr           Call Site
        fffff880`15f101c0 fffff802`b3b2d99c nt!KiSwapContext+0x76
        fffff880`15f10300 fffff802`b3b293cd nt!KiCommitThreadWait+0x23c
        fffff880`15f103c0 fffff802`b3eca2ac nt!KeWaitForMultipleObjects+0x25d
        fffff880`15f10470 fffff802`b3eca723 nt!ObWaitForMultipleObjects+0x29c
        fffff880`15f10980 fffff802`b3b02d53 nt!NtWaitForMultipleObjects+0xe3
        fffff880`15f10bd0 000007fe`f7ec319b nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @ fffff880`15f10c40)
        000000f0`72a2f428 000007fe`f4fd12c6 ntdll!NtWaitForMultipleObjects+0xa
        000000f0`72a2f430 000007fe`f6011292 KERNELBASE!WaitForMultipleObjectsEx+0xe5
        000000f0`72a2f710 000007f7`70e07c1b KERNEL32!WaitForMultipleObjects+0x12
        000000f0`72a2f750 000007fe`f601167e taskmgr!WdcServiceCache::s_InformClientsThread+0x37
        000000f0`72a2f790 000007fe`f7ee3501 KERNEL32!BaseThreadInitThunk+0x1a
        000000f0`72a2f7c0 00000000`00000000 ntdll!RtlUserThreadStart+0x1d

        THREAD fffffa8002198b00  Cid 0ca0.0aa4  Teb: 000007f770a36000 Win32Thread: 0000000000000000 WAIT: (WrQueue)
UserMode Alertable
            fffffa8003798d80  QueueObject
        Not impersonating
        DeviceMap                 fffff8a007e2e6a0
        Owning Process            fffffa8002d74180       Image:         Taskmgr.exe
        Attached Process          N/A            Image:         N/A
        Wait Start TickCount      15715946       Ticks: 25182 (0:00:06:32.841)
        Context Switch Count      3              IdealProcessor: 0
        UserTime                  00:00:00.000
        KernelTime                00:00:00.000
        Win32 Start Address ntdll!TppWorkerThread (0x000007fef7ee38c0)
        Stack Init fffff880160eddd0 Current fffff880160ed760
        Base fffff880160ee000 Limit fffff880160e8000 Call 0000000000000000
        Priority 8 BasePriority 8 PriorityDecrement 0 IoPriority 2 PagePriority 5
        Kernel stack not resident.
        Child-SP          RetAddr           Call Site
        fffff880`160ed7a0 fffff802`b3b2d99c nt!KiSwapContext+0x76
        fffff880`160ed8e0 fffff802`b3b38ddb nt!KiCommitThreadWait+0x23c
        fffff880`160ed9a0 fffff802`b3ed0b6c nt!KeRemoveQueueEx+0x26b
        fffff880`160eda50 fffff802`b3b434d5 nt!IoRemoveIoCompletion+0x4c
        fffff880`160edae0 fffff802`b3b02d53 nt!NtWaitForWorkViaWorkerFactory+0x295
        fffff880`160edc40 000007fe`f7ec46ab nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @ fffff880`160edc40)
        000000f0`77f5f608 000007fe`f7ec84b3 ntdll!ZwWaitForWorkViaWorkerFactory+0xa
        000000f0`77f5f610 000007fe`f601167e ntdll!TppWorkerThread+0x275
        000000f0`77f5f8b0 000007fe`f7ee3501 KERNEL32!BaseThreadInitThunk+0x1a
        000000f0`77f5f8e0 00000000`00000000 ntdll!RtlUserThreadStart+0x1d

        THREAD fffffa8001f3b080  Cid 0ca0.0d2c  Teb: 000007f770a4e000 Win32Thread: fffff90103f2ab90 WAIT:
(UserRequest) UserMode Non-Alertable
            fffffa80040e0220  SynchronizationEvent
            fffffa8003da2630  SynchronizationEvent
        Not impersonating
        DeviceMap                 fffff8a007e2e6a0
        Owning Process            fffffa8002d74180       Image:         Taskmgr.exe
        Attached Process          N/A            Image:         N/A
        Wait Start TickCount      15741108       Ticks: 20 (0:00:00:00.312)
        Context Switch Count      2113           IdealProcessor: 0
        UserTime                  00:00:00.000
        KernelTime                00:00:00.000
        Win32 Start Address taskmgr!WdcProcessMonitor::HangDetectionThread (0x000007f770e01354)
        Stack Init fffff88016222dd0 Current fffff88016222180
        Base fffff88016223000 Limit fffff8801621d000 Call 0000000000000000
        Priority 11 BasePriority 8 PriorityDecrement 2 IoPriority 2 PagePriority 5
        Child-SP          RetAddr           Call Site
        fffff880`162221c0 fffff802`b3b2d99c nt!KiSwapContext+0x76
        fffff880`16222300 fffff802`b3b293cd nt!KiCommitThreadWait+0x23c
        fffff880`162223c0 fffff802`b3eca2ac nt!KeWaitForMultipleObjects+0x25d
        fffff880`16222470 fffff802`b3eca723 nt!ObWaitForMultipleObjects+0x29c
        fffff880`16222980 fffff802`b3b02d53 nt!NtWaitForMultipleObjects+0xe3
        fffff880`16222bd0 000007fe`f7ec319b nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @ fffff880`16222c40)
        000000f0`72ddf648 000007fe`f4fd12c6 ntdll!NtWaitForMultipleObjects+0xa
        000000f0`72ddf650 000007fe`f6011292 KERNELBASE!WaitForMultipleObjectsEx+0xe5
        000000f0`72ddf930 000007f7`70e01398 KERNEL32!WaitForMultipleObjects+0x12
        000000f0`72ddf970 000007fe`f601167e taskmgr!WdcProcessMonitor::HangDetectionThread+0x44
        000000f0`72ddf9b0 000007fe`f7ee3501 KERNEL32!BaseThreadInitThunk+0x1a
        000000f0`72ddf9e0 00000000`00000000 ntdll!RtlUserThreadStart+0x1d

        THREAD fffffa8003bbdb00  Cid 0ca0.0ae8  Teb: 000007f770a3a000 Win32Thread: fffff90103f6e530 WAIT: (WrQueue)
UserMode Alertable
            fffffa8001e75ec0  QueueObject
        Not impersonating
        DeviceMap                 fffff8a007e2e6a0
        Owning Process            fffffa8002d74180       Image:         Taskmgr.exe
        Attached Process          N/A            Image:         N/A
        Wait Start TickCount      15741108       Ticks: 20 (0:00:00:00.312)
        Context Switch Count      7261           IdealProcessor: 0
        UserTime                  00:00:00.031
        KernelTime                00:00:00.015
        Win32 Start Address ntdll!TppWorkerThread (0x000007fef7ee38c0)
        Stack Init fffff880150c3dd0 Current fffff880150c3760
        Base fffff880150c4000 Limit fffff880150be000 Call 0000000000000000
        Priority 8 BasePriority 8 PriorityDecrement 0 IoPriority 2 PagePriority 5
        Child-SP          RetAddr           Call Site
        fffff880`150c37a0 fffff802`b3b2d99c nt!KiSwapContext+0x76
        fffff880`150c38e0 fffff802`b3b38ddb nt!KiCommitThreadWait+0x23c
        fffff880`150c39a0 fffff802`b3ed0b6c nt!KeRemoveQueueEx+0x26b
        fffff880`150c3a50 fffff802`b3b434d5 nt!IoRemoveIoCompletion+0x4c
        fffff880`150c3ae0 fffff802`b3b02d53 nt!NtWaitForWorkViaWorkerFactory+0x295
        fffff880`150c3c40 000007fe`f7ec46ab nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @ fffff880`150c3c40)
        000000f0`0010fbd8 000007fe`f7ec84b3 ntdll!ZwWaitForWorkViaWorkerFactory+0xa
        000000f0`0010fbe0 000007fe`f601167e ntdll!TppWorkerThread+0x275
        000000f0`0010fe80 000007fe`f7ee3501 KERNEL32!BaseThreadInitThunk+0x1a
        000000f0`0010feb0 00000000`00000000 ntdll!RtlUserThreadStart+0x1d

        THREAD fffffa8001e74b00  Cid 0ca0.0c34  Teb: 000007f770a34000 Win32Thread: 0000000000000000 WAIT:
(UserRequest) UserMode Non-Alertable
            fffffa8003e58460  SynchronizationTimer
        Not impersonating
        DeviceMap                 fffff8a007e2e6a0
        Owning Process            fffffa8002d74180       Image:         Taskmgr.exe
        Attached Process          N/A            Image:         N/A
        Wait Start TickCount      15740965       Ticks: 163 (0:00:00:02.542)
        Context Switch Count      10             IdealProcessor: 1
        UserTime                  00:00:00.000
        KernelTime                00:00:00.000
        Win32 Start Address combase!CRpcThreadCache::RpcWorkerThreadEntry (0x000007fef7b323a8)
        Stack Init fffff880173bedd0 Current fffff880173be0f0
        Base fffff880173bf000 Limit fffff880173b9000 Call 0000000000000000
        Priority 10 BasePriority 8 PriorityDecrement 2 IoPriority 2 PagePriority 5
        Child-SP          RetAddr           Call Site
        fffff880`173be130 fffff802`b3b2d99c nt!KiSwapContext+0x76
        fffff880`173be270 fffff802`b3b29c1f nt!KiCommitThreadWait+0x23c
        fffff880`173be330 fffff802`b3b2943e nt!KeWaitForSingleObject+0x1cf
        fffff880`173be3c0 fffff802`b3eca2ac nt!KeWaitForMultipleObjects+0x2ce
        fffff880`173be470 fffff802`b3eca723 nt!ObWaitForMultipleObjects+0x29c
        fffff880`173be980 fffff802`b3b02d53 nt!NtWaitForMultipleObjects+0xe3
        fffff880`173bebd0 000007fe`f7ec319b nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @ fffff880`173bec40)
        000000f0`0028f418 000007fe`f4fd12c6 ntdll!NtWaitForMultipleObjects+0xa
        000000f0`0028f420 000007fe`f7b3196a KERNELBASE!WaitForMultipleObjectsEx+0xe5
        000000f0`0028f700 000007fe`f7b31a03 combase!WaitCoalesced+0x96
        000000f0`0028f950 000007fe`f7b32218 combase!CROIDTable::WorkerThreadLoop+0x63
        000000f0`0028f9a0 000007fe`f7b3241f combase!CRpcThread::WorkerLoop+0x48
        000000f0`0028fc10 000007fe`f601167e combase!CRpcThreadCache::RpcWorkerThreadEntry+0x73
        000000f0`0028fc40 000007fe`f7ee3501 KERNEL32!BaseThreadInitThunk+0x1a
        000000f0`0028fc70 00000000`00000000 ntdll!RtlUserThreadStart+0x1d

        THREAD fffffa80020b5900  Cid 0ca0.0154  Teb: 000007f770a40000 Win32Thread: 0000000000000000 WAIT: (WrQueue)
UserMode Alertable
            fffffa8001e75ec0  QueueObject
        Not impersonating
        DeviceMap                 fffff8a007e2e6a0
        Owning Process            fffffa8002d74180       Image:         Taskmgr.exe
        Attached Process          N/A            Image:         N/A
        Wait Start TickCount      15740913       Ticks: 215 (0:00:00:03.354)
        Context Switch Count      6              IdealProcessor: 1
        UserTime                  00:00:00.000
        KernelTime                00:00:00.000
        Win32 Start Address ntdll!TppWorkerThread (0x000007fef7ee38c0)
        Stack Init fffff88014e29dd0 Current fffff88014e29760
        Base fffff88014e2a000 Limit fffff88014e24000 Call 0000000000000000
        Priority 8 BasePriority 8 PriorityDecrement 0 IoPriority 2 PagePriority 5
        Child-SP          RetAddr           Call Site
        fffff880`14e297a0 fffff802`b3b2d99c nt!KiSwapContext+0x76
        fffff880`14e298e0 fffff802`b3b38ddb nt!KiCommitThreadWait+0x23c
        fffff880`14e299a0 fffff802`b3ed0b6c nt!KeRemoveQueueEx+0x26b
        fffff880`14e29a50 fffff802`b3b434d5 nt!IoRemoveIoCompletion+0x4c
        fffff880`14e29ae0 fffff802`b3b02d53 nt!NtWaitForWorkViaWorkerFactory+0x295
        fffff880`14e29c40 000007fe`f7ec46ab nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @ fffff880`14e29c40)
        000000f0`0018fc78 000007fe`f7ec84b3 ntdll!ZwWaitForWorkViaWorkerFactory+0xa
        000000f0`0018fc80 000007fe`f601167e ntdll!TppWorkerThread+0x275
        000000f0`0018ff20 000007fe`f7ee3501 KERNEL32!BaseThreadInitThunk+0x1a
        000000f0`0018ff50 00000000`00000000 ntdll!RtlUserThreadStart+0x1d
```

10. Let's now check the current CPU IDT:

```
0: kd> !pcr
KPCR for Processor 0 at fffff802b3d7f000:
    Major 1 Minor 1
        NtTib.ExceptionList: fffff802b30b8000
            NtTib.StackBase: fffff802b30b9080
           NtTib.StackLimit: 000000f06e86f3e8
         NtTib.SubSystemTib: fffff802b3d7f000
              NtTib.Version: 00000000b3d7f180
          NtTib.UserPointer: fffff802b3d7f7f0
              NtTib.SelfTib: 000007f770b7d000

                    SelfPcr: 0000000000000000
                       Prcb: fffff802b3d7f180
                       Irql: 0000000000000000
                        IRR: 0000000000000000
                        IDR: 0000000000000000
              InterruptMode: 0000000000000000
                        IDT: 0000000000000000
                        GDT: 0000000000000000
                        TSS: 0000000000000000

              CurrentThread: fffffa8003db4740
```

```
                NextThread: 0000000000000000
                IdleThread: fffff802b3dd9880

                 DpcQueue:
```

If you like structure format you can use **dt** command:

```
0: kd> dt nt!_KPCR fffff802b3d7f000
   +0x000 NtTib            : _NT_TIB
   +0x000 GdtBase          : 0xfffff802`b30b8000 _KGDTENTRY64
   +0x008 TssBase          : 0xfffff802`b30b9080 _KTSS64
   +0x010 UserRsp          : 0x000000f0`6e86f3e8
   +0x018 Self             : 0xfffff802`b3d7f000 _KPCR
   +0x020 CurrentPrcb      : 0xfffff802`b3d7f180 _KPRCB
   +0x028 LockArray        : 0xfffff802`b3d7f7f0 _KSPIN_LOCK_QUEUE
   +0x030 Used_Self        : 0x000007f7`70b7d000 Void
   +0x038 IdtBase          : 0xfffff802`b30b8080 _KIDTENTRY64
   +0x040 Unused           : [2] 0
   +0x050 Irql             : 0 ''
   +0x051 SecondLevelCacheAssociativity : 0x10 ''
   +0x052 ObsoleteNumber   : 0 ''
   +0x053 Fill0            : 0 ''
   +0x054 Unused0          : [3] 0
   +0x060 MajorVersion     : 1
   +0x062 MinorVersion     : 1
   +0x064 StallScaleFactor : 0x63c
   +0x068 Unused1          : [3] (null)
   +0x080 KernelReserved   : [15] 0
   +0x0bc SecondLevelCacheSize : 0x400000
   +0x0c0 HalReserved      : [16] 0x5f217c30
   +0x100 Unused2          : 0
   +0x108 KdVersionBlock   : (null)
   +0x110 Unused3          : (null)
   +0x118 PcrAlign1        : [24] 0
   +0x180 Prcb             : _KPRCB

0: kd> !prcb
PRCB for Processor 0 at fffff802b3d7f180:
Current IRQL -- 0
Threads--  Current fffffa8003db4740 Next 0000000000000000 Idle fffff802b3dd9880
Processor Index 0 Number (0, 0) GroupSetMember 1
Interrupt Count -- 00146891
Times -- Dpc    0000026d Interrupt 00000159
         Kernel 0001cc95 User      00002a1d

0: kd> dt nt!_KPRCB  fffff802b3d7f180
   +0x000 MxCsr            : 0x1f80
   +0x004 LegacyNumber     : 0 ''
   +0x005 ReservedMustBeZero : 0 ''
   +0x006 InterruptRequest : 0 ''
   +0x007 IdleHalt         : 0 ''
   +0x008 CurrentThread    : 0xfffffa80`03db4740 _KTHREAD
   +0x010 NextThread       : (null)
   +0x018 IdleThread       : 0xfffff802`b3dd9880 _KTHREAD
   +0x020 NestingLevel     : 0 ''
   +0x021 ClockOwner       : 0x1 ''
   +0x022 PendingTick      : 0 ''
   +0x023 PrcbPad00        : [1]  ""
   +0x024 Number           : 0
   +0x028 RspBase          : 0xfffff880`15925dd0
   +0x030 PrcbLock         : 0
   +0x038 PrcbPad01        : 0
   +0x040 ProcessorState   : _KPROCESSOR_STATE
   +0x5f0 CpuType          : 6 ''
   +0x5f1 CpuID            : 1 ''
   +0x5f2 CpuStep          : 0xf0b
   +0x5f2 CpuStepping      : 0xb ''
   +0x5f3 CpuModel         : 0xf ''
   +0x5f4 MHz              : 0x63c
   +0x5f8 HalReserved      : [8] 0
   +0x638 MinorVersion     : 1
   +0x63a MajorVersion     : 1
   +0x63c BuildType        : 0 ''
   +0x63d CpuVendor        : 0x2 ''
   +0x63e CoresPerPhysicalProcessor : 0x2 ''
   +0x63f LogicalProcessorsPerCore : 0x1 ''
   +0x640 ApicMask         : 0xfffffffe
   +0x644 CFlushSize       : 0x40
   +0x648 AcpiReserved     : (null)
   +0x650 InitialApicId    : 0
   +0x654 Stride           : 2
   +0x658 Group            : 0
   +0x660 GroupSetMember   : 1
   +0x668 GroupIndex       : 0 ''
   +0x670 LockQueue        : [17] _KSPIN_LOCK_QUEUE
   +0x780 PPLookasideList  : [16] _PP_LOOKASIDE_LIST
   +0x880 PPNxPagedLookasideList : [32] _GENERAL_LOOKASIDE_POOL
   +0x1480 PPNPagedLookasideList : [32] _GENERAL_LOOKASIDE_POOL
   +0x2080 PPPagedLookasideList : [32] _GENERAL_LOOKASIDE_POOL
   +0x2c80 PrcbPad20        : 0
   +0x2c88 DeferredReadyListHead : _SINGLE_LIST_ENTRY
   +0x2c90 MmPageFaultCount : 0n1729599
   +0x2c94 MmCopyOnWriteCount : 0n27918
   +0x2c98 MmTransitionCount : 0n593150
   +0x2c9c MmDemandZeroCount : 0n882660
   +0x2ca0 MmPageReadCount  : 0n382444
   +0x2ca4 MmPageReadIoCount : 0n57376
   +0x2ca8 MmDirtyPagesWriteCount : 0n35128
   +0x2cac MmDirtyWriteIoCount : 0n582
   +0x2cb0 MmMappedPagesWriteCount : 0n178
   +0x2cb4 MmMappedWriteIoCount : 0n15
   +0x2cb8 KeSystemCalls    : 0x20f77d0
   +0x2cbc KeContextSwitches : 0x1aecf6
   +0x2cc0 CcFastReadNoWait : 0
   +0x2cc4 CcFastReadWait   : 0x6850
   +0x2cc8 CcFastReadNotPossible : 0x32
   +0x2ccc CcCopyReadNoWait : 0
   +0x2cd0 CcCopyReadWait   : 0x7793
   +0x2cd4 CcCopyReadNoWaitMiss : 0
   +0x2cd8 LookasideIrpFloat : 0n2147483647
   +0x2cdc IoReadOperationCount : 0n50462
   +0x2ce0 IoWriteOperationCount : 0n56714
   +0x2ce4 IoOtherOperationCount : 0n323985
   +0x2ce8 IoReadTransferCount : _LARGE_INTEGER 0x1e1e96d6
   +0x2cf0 IoWriteTransferCount : _LARGE_INTEGER 0x2168e9a3
   +0x2cf8 IoOtherTransferCount : _LARGE_INTEGER 0x1335cfe
   +0x2d00 PacketBarrier    : 0n0
   +0x2d04 TargetCount      : 0n0
   +0x2d08 IpiFrozen        : 0
   +0x2d0c PrcbPad40        : [29] 0
   +0x2d80 DpcData          : [2] _KDPC_DATA
   +0x2dc0 DpcStack         : 0xfffff802`b30c5fb0 Void
   +0x2dc8 MaximumDpcQueueDepth : 0n4
   +0x2dcc DpcRequestRate   : 8
   +0x2dd0 MinimumDpcRate   : 3
   +0x2dd4 DpcLastCount     : 0x5c62b
   +0x2dd8 ThreadDpcEnable  : 0x1 ''
   +0x2dd9 QuantumEnd       : 0 ''
   +0x2dda DpcRoutineActive : 0 ''
   +0x2ddb IdleSchedule     : 0 ''
   +0x2ddc DpcRequestSummary : 0n0
   +0x2ddc DpcRequestSlot   : [2] 0n0
   +0x2ddc NormalDpcState   : 0n0
   +0x2dde ThreadDpcState   : 0n0
   +0x2ddc DpcNormalProcessingActive : 0y0
   +0x2ddc DpcNormalProcessingRequested : 0y0
   +0x2ddc DpcNormalThreadSignal : 0y0
   +0x2ddc DpcNormalTimerExpiration : 0y0
   +0x2ddc DpcNormalDpcPresent : 0y0
   +0x2ddc DpcNormalLocalInterrupt : 0y0
   +0x2ddc DpcNormalSpare   : 0y0000000000 (0)
   +0x2ddc DpcThreadActive  : 0y0
   +0x2ddc DpcThreadRequested : 0y0
   +0x2ddc DpcThreadSpare   : 0y00000000000000 (0)
   +0x2de0 LastTimerHand    : 0x8eefc3
   +0x2de4 LastTick         : 0xf030c8
   +0x2de8 ClockInterrupts  : 0x1e7f4
   +0x2dec ReadyScanTick    : 0xf03113
   +0x2df0 BalanceState     : 0 ''
   +0x2df1 PrcbPad50        : [7]  ""
   +0x2df8 InterruptLastCount : 0x146853
   +0x2dfc InterruptRate    : 3
   +0x2e00 TimerTable       : _KTIMER_TABLE
   +0x5000 DpcGate          : _KGATE
   +0x5018 PrcbPad52        : (null)
   +0x5020 CallDpc          : _KDPC
   +0x5060 ClockKeepAlive   : 0n1
   +0x5064 PrcbPad60        : [2]  ""
   +0x5066 NmiActive        : 0
   +0x5068 DpcWatchdogPeriod : 0n1924
   +0x506c DpcWatchdogCount : 0n1918
   +0x5070 KeSpinLockOrdering : 0n0
   +0x5074 PrcbPad70        : [1] 0
   +0x5078 CachedPtes       : (null)
   +0x5080 WaitListHead     : _LIST_ENTRY [ 0xfffffa80`01e03158 - 0xfffffa80`0419abd8 ]
   +0x5090 WaitLock         : 0
   +0x5098 ReadySummary     : 0x1000
   +0x509c QueueIndex       : 1
   +0x50a0 ReadyQueueWeight : 0xc
   +0x50a4 PrcbPad75        : 0
   +0x50a8 TimerExpirationDpc : _KDPC
   +0x50e8 BuddyPrcb        : (null)
   +0x50f0 ScbQueue         : _RTL_RB_TREE
   +0x5100 DispatcherReadyListHead : [32] _LIST_ENTRY [ 0xfffff802`b3d84280 -
0xfffff802`b3d84280 ]
   +0x5300 InterruptCount   : 0x146891
   +0x5304 KernelTime       : 0x1cc95
   +0x5308 UserTime         : 0x2a1d
   +0x530c DpcTime          : 0x26d
   +0x5310 InterruptTime    : 0x159
   +0x5314 AdjustDpcThreshold : 2
   +0x5318 DebuggerSavedIRQL : 0 ''
   +0x5319 GroupSchedulingOverQuota : 0 ''
   +0x531a DeepSleep        : 0 ''
   +0x531b PrcbPad80        : [1]  ""
   +0x531c ScbOffset        : 0x40
   +0x5320 DpcTimeCount     : 0
   +0x5324 DpcTimeLimit     : 0x282
   +0x5328 PeriodicCount    : 0
   +0x532c PeriodicBias     : 0
   +0x5330 AvailableTime    : 0xc07
   +0x5334 KeExceptionDispatchCount : 0x324
   +0x5338 ParentNode       : 0xfffff802`b3d0d000 _KNODE
   +0x5340 StartCycles      : 0x0000020d`2f5acf08
   +0x5348 GenerationTarget : 0x2431db
   +0x5350 AffinitizedCycles : 0x00000004`0f38cfd0
   +0x5358 PrcbPad81        : 0
   +0x5360 MmSpinLockOrdering : 0n0
   +0x5364 PageColor        : 0x498b
   +0x5368 NodeColor        : 0
   +0x536c NodeShiftedColor : 0
   +0x5370 SecondaryColorMask : 0x3f
   +0x5374 PrcbPad83        : 0
   +0x5378 CycleTime        : 0x00000007`8a855d30
   +0x5380 CcFastMdlReadNoWait : 0
   +0x5384 CcFastMdlReadWait : 0
   +0x5388 CcFastMdlReadNotPossible : 0
   +0x538c CcMapDataNoWait  : 0
   +0x5390 CcMapDataWait    : 0x468c4
   +0x5394 CcPinMappedDataCount : 0xa006
   +0x5398 CcPinReadNoWait  : 2
   +0x539c CcPinReadWait    : 0x3cd4
   +0x53a0 CcMdlReadNoWait  : 0
   +0x53a4 CcMdlReadWait    : 0x32
   +0x53a8 CcLazyWriteHotSpots : 0x76
   +0x53ac CcLazyWriteIos   : 0xb75
   +0x53b0 CcLazyWritePages : 0x2692c
   +0x53b4 CcDataFlushes    : 0x1c52
   +0x53b8 CcDataPages      : 0x309c2
   +0x53bc CcLostDelayedWrites : 0
   +0x53c0 CcFastReadResourceMiss : 0
   +0x53c4 CcCopyReadWaitMiss : 0xd84c
   +0x53c8 CcFastMdlReadResourceMiss : 0
   +0x53cc CcMapDataNoWaitMiss : 0
   +0x53d0 CcMapDataWaitMiss : 0xead
   +0x53d4 CcPinReadNoWaitMiss : 0
   +0x53d8 CcPinReadWaitMiss : 0x148
   +0x53dc CcMdlReadNoWaitMiss : 0
   +0x53e0 CcMdlReadWaitMiss : 0
   +0x53e4 CcReadAheadIos   : 0x111d
   +0x53e8 MmCacheTransitionCount : 0n0
   +0x53ec MmCacheReadCount : 0n0
   +0x53f0 MmCacheIoCount   : 0n0
   +0x53f4 PrcbPad91        : [3] 0
   +0x5400 PowerState       : _PROCESSOR_POWER_STATE
   +0x55c8 ScbList          : _LIST_ENTRY [ 0xfffffa80`030575f0 - 0xfffffa80`036ab930 ]
   +0x55d8 PrcbPad92        : [22] 0
   +0x5630 KeAlignmentFixupCount : 0
   +0x5638 DpcWatchdogDpc   : _KDPC
   +0x5678 DpcWatchdogTimer : _KTIMER
   +0x56b8 Cache            : [5] _CACHE_DESCRIPTOR
   +0x56f4 CacheCount       : 3
   +0x56f8 CachedCommit     : 0xfe
   +0x56fc CachedResidentAvailable : 0x91
   +0x5700 HyperPte         : 0xfffff880`00800005 Void
   +0x5708 WheaInfo         : 0xfffffa80`0182d7c0 Void
   +0x5710 EtwSupport       : 0xfffffa80`01815010 Void
   +0x5720 InterruptObjectPool : _SLIST_HEADER
   +0x5730 HypercallPageList : _SLIST_HEADER
   +0x5740 HypercallPageVirtual : (null)
   +0x5748 VirtualApicAssist : (null)
   +0x5750 StatisticsPage   : (null)
   +0x5758 PackageProcessorSet : _KAFFINITY_EX
   +0x5800 CacheProcessorMask : [5] 1
   +0x5828 ScanSiblingMask  : 3
   +0x5830 ScanSiblingIndex : 0
   +0x5834 LLCLevel         : 2
   +0x5838 CoreProcessorSet : 1
   +0x5840 ProcessorProfileControlArea : (null)
   +0x5848 ProfileEventIndexAddress : 0xfffff802`b3d849c8 Void
   +0x5850 PrcbPad94        : [6] 0
   +0x5880 SynchCounters    : _SYNCH_COUNTERS
   +0x5938 FsCounters       : _FILESYSTEM_DISK_COUNTERS
   +0x5948 VendorString     : [13]  "GenuineIntel"
   +0x5955 PrcbPad10        : [3]  ""
   +0x5958 FeatureBits      : 0x291b3ffe
   +0x5960 UpdateSignature  : _LARGE_INTEGER 0x000000ba`00000000
   +0x5968 Context          : 0xfffff802`b3d7f2a0 _CONTEXT
   +0x5970 ContextFlagsInit : 0x10000b
   +0x5978 ExtendedState    : (null)
   +0x5980 EntropyTimingState : _KENTROPY_TIMING_STATE
   +0x5b00 Mailbox          : (null)
   +0x5b40 RequestMailbox   : [1] _REQUEST_MAILBOX

0: kd> !idt

Dumping IDT: fffff802b30b8080

00:	fffff802b3b00440 nt!KiDivideErrorFault
01:	fffff802b3b00540 nt!KiDebugTrapOrFault
02:	fffff802b3b00700 nt!KiNmiInterrupt	Stack = 0xFFFFF802B30CA000
03:	fffff802b3b00a80 nt!KiBreakpointTrap
04:	fffff802b3b00b80 nt!KiOverflowTrap
05:	fffff802b3b00c80 nt!KiBoundFault
06:	fffff802b3b00d80 nt!KiInvalidOpcodeFault
07:	fffff802b3b00fc0 nt!KiNpxNotAvailableFault
08:	fffff802b3b01080 nt!KiDoubleFaultAbort	Stack = 0xFFFFF802B30C8000
09:	fffff802b3b01140 nt!KiNpxSegmentOverrunAbort
0a:	fffff802b3b01200 nt!KiInvalidTssFault
0b:	fffff802b3b012c0 nt!KiSegmentNotPresentFault
0c:	fffff802b3b01400 nt!KiStackFault
0d:	fffff802b3b01540 nt!KiGeneralProtectionFault
0e:	fffff802b3b01680 nt!KiPageFault
10:	fffff802b3b01a40 nt!KiFloatingErrorFault
11:	fffff802b3b01bc0 nt!KiAlignmentFault
12:	fffff802b3b01cc0 nt!KiMcheckAbort	Stack = 0xFFFFF802B30CC000
13:	fffff802b3b02340 nt!KiXmmException
1f:	fffff802b3b65ad0 nt!KiApcInterrupt
29:	fffff802b3b02500 nt!KiRaiseSecurityCheckFailure
2c:	fffff802b3b02600 nt!KiRaiseAssertion
```

```
2d:    fffff802b3b02700 nt!KiDebugServiceTrap
2f:    fffff802b3bc5190 nt!KiDpcInterrupt
30:    fffff802b3afb6d0 nt!KiHvInterrupt
31:    fffff802b3afba20 nt!KiVmbusInterrupt0
32:    fffff802b3afbd60 nt!KiVmbusInterrupt1
33:    fffff802b3afc0a0 nt!KiVmbusInterrupt2
34:    fffff802b3afc3e0 nt!KiVmbusInterrupt3
37:    fffff802b3a69560 hal!HalpInterruptSpuriousService (KINTERRUPT fffff802b3a694d0)

3f:    fffff802b3a691f0 hal!HalpInterruptSpuriousService (KINTERRUPT fffff802b3a69160)

50:    fffff802b3a69090 hal!HalpInterruptCmciService (KINTERRUPT fffff802b3a69000)

60:    fffff88000993ed0 pci!ExpressRootPortMessageRoutine (KINTERRUPT fffff88000993e40)

71:    fffff88000993990 USBPORT!USBPORT_InterruptService (KINTERRUPT fffff88000993900)

                        USBPORT!USBPORT_InterruptService (KINTERRUPT fffff88000993780)

                        Unable to load image \SystemRoot\system32\DRIVERS\bcmwl63a.sys, Win32
error 0n2
bcmwl63a!wl_isr60 (NDIS) (KINTERRUPT fffff880009936c0)

                        dxgkrnl!DpiFdoLineInterruptRoutine (KINTERRUPT fffff88000993300)

81:    fffff88000993b10 USBPORT!USBPORT_InterruptService (KINTERRUPT fffff88000993a80)

                        USBPORT!USBPORT_InterruptService (KINTERRUPT fffff880009933c0)

                        HDAudBus!HdaController::Isr (KINTERRUPT fffff88000993600)

91:    fffff88000993c90 ataport!IdePortInterrupt (KINTERRUPT fffff88000993c00)

                        ataport!IdePortInterrupt (KINTERRUPT fffff88000993b40)

                        USBPORT!USBPORT_InterruptService (KINTERRUPT fffff88000993540)

a1:    fffff88000993a50 ataport!IdePortInterrupt (KINTERRUPT fffff880009939c0)

                        ataport!IdePortInterrupt (KINTERRUPT fffff88000993cc0)

                        USBPORT!USBPORT_InterruptService (KINTERRUPT fffff88000993840)

                        USBPORT!USBPORT_InterruptService (KINTERRUPT fffff88000993480)

b0:    fffff88000993f90 ACPI!ACPIInterruptServiceRoutine (KINTERRUPT fffff88000993f00)

b1:    fffff88000993e10 pci!ExpressRootPortMessageRoutine (KINTERRUPT fffff88000993d80)

c0:    fffff802b3a692a0 hal!HalpInterruptStubService (KINTERRUPT fffff802b3a69210)

c2:    fffff802b3a696c0 hal!HalpDmaControllerInterruptRoutine (KINTERRUPT fffff802b3a69630)

d1:    fffff802b3a69610 hal!HalpTimerClockInterrupt (KINTERRUPT fffff802b3a69580)

df:    fffff802b3a69400 hal!HalpInterruptRebootService (KINTERRUPT fffff802b3a69370)

e1:    fffff802b3b30f10 nt!KiIpiInterrupt
e2:    fffff802b3a69350 hal!HalpInterruptLocalErrorService (KINTERRUPT fffff802b3a692c0)
```

```
e3:    fffff802b3a69140 hal!HalpInterruptDeferredRecoveryService (KINTERRUPT fffff802b3a690b0)

fe:    fffff802b3a694b0 hal!HalpPerfInterrupt (KINTERRUPT fffff802b3a69420)

0: kd> !idt -a

Dumping IDT: fffff802b30b8080

00:    fffff802b3b00440 nt!KiDivideErrorFault
01:    fffff802b3b00540 nt!KiDebugTrapOrFault
02:    fffff802b3b00700 nt!KiNmiInterrupt     Stack = 0xFFFFF802B30CA000
03:    fffff802b3b00a80 nt!KiBreakpointTrap
04:    fffff802b3b00b80 nt!KiOverflowTrap
05:    fffff802b3b00c80 nt!KiBoundFault
06:    fffff802b3b00d80 nt!KiInvalidOpcodeFault
07:    fffff802b3b00fc0 nt!KiNpxNotAvailableFault
08:    fffff802b3b01080 nt!KiDoubleFaultAbort Stack = 0xFFFFF802B30C8000
09:    fffff802b3b01140 nt!KiNpxSegmentOverrunAbort
0a:    fffff802b3b01200 nt!KiInvalidTssFault
0b:    fffff802b3b012c0 nt!KiSegmentNotPresentFault
0c:    fffff802b3b01400 nt!KiStackFault
0d:    fffff802b3b01540 nt!KiGeneralProtectionFault
0e:    fffff802b3b01680 nt!KiPageFault
0f:    fffff802b3cfa0f0 nt!KxUnexpectedInterrupt0+0xF0
10:    fffff802b3b01a40 nt!KiFloatingErrorFault
11:    fffff802b3b01bc0 nt!KiAlignmentFault
12:    fffff802b3b01cc0 nt!KiMcheckAbort      Stack = 0xFFFFF802B30CC000
13:    fffff802b3b02340 nt!KiXmmException
14:    fffff802b3cfa140 nt!KxUnexpectedInterrupt0+0x140
15:    fffff802b3cfa150 nt!KxUnexpectedInterrupt0+0x150
16:    fffff802b3cfa160 nt!KxUnexpectedInterrupt0+0x160
17:    fffff802b3cfa170 nt!KxUnexpectedInterrupt0+0x170
18:    fffff802b3cfa180 nt!KxUnexpectedInterrupt0+0x180
19:    fffff802b3cfa190 nt!KxUnexpectedInterrupt0+0x190
1a:    fffff802b3cfa1a0 nt!KxUnexpectedInterrupt0+0x1A0
1b:    fffff802b3cfa1b0 nt!KxUnexpectedInterrupt0+0x1B0
1c:    fffff802b3cfa1c0 nt!KxUnexpectedInterrupt0+0x1C0
1d:    fffff802b3cfa1d0 nt!KxUnexpectedInterrupt0+0x1D0
[...]
```

Note that some interrupts have their own stack.

11. Let's now check the raw stack data for the current thread:

```
0: kd> !thread -1 3f
THREAD fffffa8003db4740  Cid 0ca0.03e0  Teb: 000007f770b7d000 Win32Thread: fffff90104094830
RUNNING on processor 0
Not impersonating
DeviceMap                 fffff8a007e2e6a0
Owning Process            fffffa8002d74180       Image:         Taskmgr.exe
Attached Process          N/A            Image:         N/A
Wait Start TickCount      15741128       Ticks: 0
Context Switch Count      31359          IdealProcessor: 0
UserTime                  00:00:09.859
KernelTime                00:00:07.394
Win32 Start Address taskmgr!wWinMainCRTStartup (0x000007f770e68688)
Stack Init fffff88015925dd0 Current fffff88015925800
Base fffff88015926000 Limit fffff88015920000 Call 0000000000000000
Priority 13 BasePriority 9 PriorityDecrement 2 IoPriority 2 PagePriority 5
```

```
Child-SP          RetAddr           Call Site
fffff880`15925ae8 fffff802`b400f0dd nt!KeBugCheckEx
fffff880`15925af0 fffff802`b3ea8f6d nt!PspCatchCriticalBreak+0xad
fffff880`15925b30 fffff802`b3ea8019 nt! ?? ::NNGAKEGL::`string'+0x46f60
fffff880`15925b90 fffff802`b3ea7e52 nt!PspTerminateProcess+0x6d
fffff880`15925bd0 fffff802`b3b02d53 nt!NtTerminateProcess+0x9e
fffff880`15925c40 000007fe`f7ec2eaa nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @
fffff880`15925c40)
000000f0`6e86f3e8 000007fe`f4ff1295 ntdll!NtTerminateProcess+0xa
000000f0`6e86f3f0 000007f7`70e012ba KERNELBASE!TerminateProcess+0x25
000000f0`6e86f420 000007f7`70df3698 taskmgr!WdcProcessMonitor::OnProcessCommand+0x1b6
000000f0`6e86f4b0 000007f7`70df55bb taskmgr!WdcListView::OnProcessCommand+0x1e0
000000f0`6e86f5a0 000007f7`70df5b47 taskmgr!WdcListView::OnCommand+0x123
000000f0`6e86f5f0 000007fe`f2227239 taskmgr!WdcListView::OnMessage+0x287
000000f0`6e86f710 000007fe`f2a82d23 DUI70!DirectUI::HWNDHost::_CtrlWndProc+0xa1
000000f0`6e86f770 000007fe`f56c171e DUser!WndBridge::RawWndProc+0x73
000000f0`6e86f7e0 000007fe`f56c14d7 USER32!UserCallWinProcCheckWow+0x13a
000000f0`6e86f8a0 000007f7`70e1b0e1 USER32!DispatchMessageWorker+0x1a7
000000f0`6e86f920 000007f7`70e685e6 taskmgr!wWinMain+0x44d
000000f0`6e86fde0 000007fe`f601167e taskmgr!CBaseRPCTimeout::Disarm+0x31a
000000f0`6e86fea0 000007fe`f7ee3501 KERNEL32!BaseThreadInitThunk+0x1a
000000f0`6e86fed0 00000000`00000000 ntdll!RtlUserThreadStart+0x1d

0: kd> dps fffff88015920000 fffff88015926000
[...]
fffff880`15924098  00000000`00000000
fffff880`159240a0  00000000`00000000
fffff880`159240a8  00000000`00000000
fffff880`159240b0  00000000`00000000
fffff880`159240b8  00000000`00000000
fffff880`159240c0  fffff880`00000000
fffff880`159240c8  fffff880`040067e4 igdkmd64!PORTCONTROLLER_EnumEnabledPortsOnPipe+0x64
fffff880`159240d0  fffff880`03cec200
fffff880`159240d8  04524320`00000048
fffff880`159240e0  00000500`000005a0
fffff880`159240e8  fffff880`03f8b652 igdkmd64!ExtInterface_ReadULONG+0x52
fffff880`159240f0  fffffa80`01a2e000
fffff880`159240f8  fffff880`03cec200
fffff880`15924100  00000320`abcd0003
fffff880`15924108  00000323`00000336
fffff880`15924110  fffff880`03cec204
fffff880`15924118  fffffa80`01a2eefc
fffff880`15924120  04524320`00000048
fffff880`15924128  fffff880`03f8b5ec igdkmd64!ExtInterface_WriteULONG+0x5c
fffff880`15924130  fffffa80`01a2e000
fffff880`15924138  fffff880`03cec204
fffff880`15924140  00000337`00000003
fffff880`15924148  00000320`00000320
fffff880`15924150  fffffa80`01a2e000
fffff880`15924158  fffffa80`01a2eefc
fffff880`15924160  00000000`0800000c
fffff880`15924168  fffff880`04033015 igdkmd64!MMIOREG_WriteValue+0x55
fffff880`15924170  fffffa80`01a6d010
fffff880`15924178  fffff880`00061204
fffff880`15924180  fffff880`00000003
fffff880`15924188  fffff880`00000000
fffff880`15924190  fffffa80`01a6d010
fffff880`15924198  00000005`00000000
fffff880`159241a0  fffff801`00000001
fffff880`159241a8  fffff880`0406efd8 igdkmd64!PORTBASE_SetEncoderRegisterValue+0x1c8
```

```
fffff880`159241b0  fffff880`159241f0
fffff880`159241b8  fffff880`00000003
fffff880`159241c0  fffff880`00000000
fffff880`159241c8  fffff880`00000000
fffff880`159241d0  fffff880`0000fffc
fffff880`159241d8  fffff880`04033270 igdkmd64!MMIOREG_WriteMaskedByteValue
fffff880`159241e0  fffff801`00000002
fffff880`159241e8  fffff880`04033320 igdkmd64!MMIOREG_Commit
fffff880`159241f0  fffff880`040330e0 igdkmd64!MMIOREG_ReadValue
fffff880`159241f8  fffff880`04033160 igdkmd64!MMIOREG_ReadByteValue
fffff880`15924200  fffff880`04032fc0 igdkmd64!MMIOREG_WriteValue
fffff880`15924208  fffff880`04033040 igdkmd64!MMIOREG_WriteByteValue
fffff880`15924210  fffff880`04033200 igdkmd64!MMIOREG_WriteMaskedValue
fffff880`15924218  fffff880`04033270 igdkmd64!MMIOREG_WriteMaskedByteValue
fffff880`15924220  fffff880`04033300 igdkmd64!MMIOREG_EnableCaching
fffff880`15924228  fffff880`04033320 igdkmd64!MMIOREG_Commit
fffff880`15924230  fffff880`04033390 igdkmd64!MMIOREG_ReadWrite
fffff880`15924238  fffff880`040333d0 igdkmd64!MMIOREG_SaveValue
fffff880`15924240  fffff880`04033410 igdkmd64!MMIOREG_RestoreValue
fffff880`15924248  fffff880`04033460 igdkmd64!MMIOREG_RestoreMaskedValue
fffff880`15924250  fffff880`04033550 igdkmd64!MMIOREG_ReadMultiValue
fffff880`15924258  fffff880`040334e0 igdkmd64!MMIOREG_WriteMultiValue
fffff880`15924260  00000000`00061204
fffff880`15924268  0000fffc`00000000
fffff880`15924270  00000000`01a6cd00
fffff880`15924278  fffff800`00061200
fffff880`15924280  fffff880`04032fb0 igdkmd64!MMIOREG_Destroy
fffff880`15924288  fffff880`03fbc52c igdkmd64!GMCHBASE_GetPortObject+0x2c
fffff880`15924290  00000000`00000028
fffff880`15924298  abcd0003`abcd0003
fffff880`159242a0  00000000`00000000
fffff880`159242a8  fffff880`03fbc404 igdkmd64!GMCHBASE_SetInternalEncoderRegister+0xb4
fffff880`159242b0  fffffa80`01a6cde0
fffff880`159242b8  fffff880`00061204
fffff880`159242c0  fffff880`00000003
fffff880`159242c8  fffff880`00000001
fffff880`159242d0  fffffa01`00000005
fffff880`159242d8  fffffa80`01a6cde0
fffff880`159242e0  00000001`00000001
fffff880`159242e8  fffffa80`01a6cde0
fffff880`159242f0  fffff880`15924388
fffff880`159242f8  fffff880`03fd1d32 igdkmd64!INTLVDSENCODER_SetTiming+0x552
fffff880`15924300  fffffa80`01a97000
fffff880`15924308  fffff880`00000001
fffff880`15924310  fffff880`00061204
fffff880`15924318  fffff880`00000003
fffff880`15924320  fffff880`159244b0
fffff880`15924328  00000000`00000000
fffff880`15924330  00000000`00000000
fffff880`15924338  00000000`00000000
fffff880`15924340  04524320`00000048
fffff880`15924348  00000500`000005a0
fffff880`15924350  0000059f`00000500
fffff880`15924358  0000054f`00000530
fffff880`15924360  00000337`0000c4ab
fffff880`15924368  00000320`00000320
fffff880`15924370  00000323`00000336
fffff880`15924378  0000003d`00000328
fffff880`15924380  00000000`0800000c
fffff880`15924388  00000000`00000000
```

```
fffff880`15924390  00000002`00000000
fffff880`15924398  00000000`00000000
fffff880`159243a0  00000000`00000000
fffff880`159243a8  00000000`00000000
fffff880`159243b0  00001000`00000000
fffff880`159243b8  00000000`00000001
fffff880`159243c0  fffffa80`01a97000
fffff880`159243c8  00000000`00000003
fffff880`159243d0  04524320`00000048
fffff880`159243d8  00000500`000005a0
fffff880`159243e0  0000059f`00000500
fffff880`159243e8  0000054f`00000530
fffff880`159243f0  00000337`0000c4ab
fffff880`159243f8  00000320`00000320
fffff880`15924400  00000323`00000336
fffff880`15924408  0000003d`00000328
fffff880`15924410  00000000`0000000c
fffff880`15924418  00000000`00000407
fffff880`15924420  04524320`00000048
fffff880`15924428  00000500`000005a0
fffff880`15924430  0000059f`00000500
fffff880`15924438  0000054f`00000530
fffff880`15924440  00000337`0000c4ab
fffff880`15924448  00000320`00000320
fffff880`15924450  00000323`00000336
fffff880`15924458  0000003d`00000328
fffff880`15924460  00000000`0800000c
fffff880`15924468  00000000`00000000
fffff880`15924470  000001e0`00000280
fffff880`15924478  00000000`00000004
fffff880`15924480  00000000`0000003c
fffff880`15924488  00000280`00000000
fffff880`15924490  0000003c`000001e0
fffff880`15924498  00000000`00000000
fffff880`159244a0  00000000`00000000
fffff880`159244a8  00000000`00000001
fffff880`159244b0  0000007f`00000000
fffff880`159244b8  0000007f`0000007f
fffff880`159244c0  00000001`00000001
fffff880`159244c8  00000000`00000005
fffff880`159244d0  00000000`00000000
fffff880`159244d8  fffff880`03f8ce7a igdkmd64!GetCSLSBIOSProtocolObject+0x3a
fffff880`159244e0  fffffa80`01a08830
fffff880`159244e8  00000000`00000000
fffff880`159244f0  00000000`00000000
fffff880`159244f8  00000000`00000000
fffff880`15924500  fffffa80`01a145f0
fffff880`15924508  fffffa80`01a06010
fffff880`15924510  000001e0`00000280
fffff880`15924518  fffff880`03fa4b0e igdkmd64!MODESMANAGER_PostSetMode+0x1e
fffff880`15924520  00000000`0000003c
fffff880`15924528  00000280`00000000
fffff880`15924530  0000003c`000001e0
fffff880`15924538  00000000`00000000
fffff880`15924540  00000000`00000000
fffff880`15924548  fffffa80`01a06010
fffff880`15924550  fffffa80`01a12e00
fffff880`15924558  fffff880`15924a4c
fffff880`15924560  fffffa80`01a151b6
fffff880`15924568  fffff880`03f975f9 igdkmd64!MODESMANAGER_SetMode+0x6b9
```

```
fffff880`15924570  fffffa80`01a15010
fffff880`15924578  fffff880`15924da0
fffff880`15924580  fffff880`15924da0
fffff880`15924588  fffff880`15924a10
fffff880`15924590  fffff880`15924a50
fffff880`15924598  fffffa80`017e0700
fffff880`159245a0  fffffa80`01a2d010
fffff880`159245a8  fffff8a0`01be49a0
fffff880`159245b0  00000000`00000000
fffff880`159245b8  0000007f`0000007f
fffff880`159245c0  00000001`0000007f
fffff880`159245c8  00000005`00000001
fffff880`159245d0  00000000`00000000
fffff880`159245d8  00000000`00000000
fffff880`159245e0  00000000`00000000
fffff880`159245e8  00000000`00000000
fffff880`159245f0  00000000`00000000
fffff880`159245f8  00000000`00000000
fffff880`15924600  00000000`00000000
fffff880`15924608  0000007f`00000000
fffff880`15924610  0000007f`0000007f
fffff880`15924618  00000001`0000007f
fffff880`15924620  00000000`00000000
fffff880`15924628  00000000`00000000
fffff880`15924630  00000000`00000000
fffff880`15924638  00000000`00000000
fffff880`15924640  00000000`00000000
fffff880`15924648  00000000`00000000
fffff880`15924650  00000000`00000000
fffff880`15924658  00000000`00000000
fffff880`15924660  fffff901`00000000
fffff880`15924668  fffff901`000d2e20
fffff880`15924670  fffff880`00000000
fffff880`15924678  fffffa80`01a10010
fffff880`15924680  00000000`00000000
fffff880`15924688  00000000`00000000
fffff880`15924690  00000000`00000000
fffff880`15924698  00000000`00000000
fffff880`159246a0  00000000`00000000
fffff880`159246a8  00000000`00000000
fffff880`159246b0  00000000`00000000
fffff880`159246b8  00000000`00000000
fffff880`159246c0  00000000`00000000
fffff880`159246c8  00000000`00000000
fffff880`159246d0  00000000`00000000
fffff880`159246d8  fffff802`b3ec289b nt!RtlAnsiCharToUnicodeChar+0x4b
fffff880`159246e0  04070400`00000201
fffff880`159246e8  fffffa80`01a97000
fffff880`159246f0  00000000`00000000
fffff880`159246f8  00000000`000007ff
fffff880`15924700  04070400`00000001
fffff880`15924708  00000000`00000000
fffff880`15924710  00000000`00000000
fffff880`15924718  00000000`00000000
fffff880`15924720  00000000`00000000
fffff880`15924728  00000000`00000000
fffff880`15924730  00000000`00000000
fffff880`15924738  00000000`00000000
fffff880`15924740  00000000`00000000
fffff880`15924748  00000000`00000000
```

```
fffff880`15924750  00000000`00000000
fffff880`15924758  00000000`00000000
fffff880`15924760  00000000`00000000
fffff880`15924768  00000000`00000000
fffff880`15924770  00000000`00000000
fffff880`15924778  00000000`00000000
fffff880`15924780  00000000`00000000
fffff880`15924788  fffff880`ffffff00
fffff880`15924790  000001e0`00000280
fffff880`15924798  00000000`00000004
fffff880`159247a0  00000000`0000003c
fffff880`159247a8  00000280`00000000
fffff880`159247b0  0000003c`000001e0
fffff880`159247b8  00000000`00000000
fffff880`159247c0  00000000`00000000
fffff880`159247c8  fffff880`00000001
fffff880`159247d0  ffffffff`00000001
fffff880`159247d8  00000000`00000000
fffff880`159247e0  04000000`00010001
fffff880`159247e8  00000000`00000407
fffff880`159247f0  00000000`00000000
fffff880`159247f8  00000000`00000000
fffff880`15924800  00000000`00000000
fffff880`15924808  00000000`00000000
fffff880`15924810  00000000`00000000
fffff880`15924818  00000000`00000000
fffff880`15924820  00000000`00000000
fffff880`15924828  00000000`00000000
fffff880`15924830  00000000`00000000
fffff880`15924838  00000000`00000000
fffff880`15924840  00000000`00000000
fffff880`15924848  00000000`00000000
fffff880`15924850  00000000`00000000
fffff880`15924858  00000000`00000000
fffff880`15924860  00000000`00000000
fffff880`15924868  00000000`00000000
fffff880`15924870  00000000`00ffff00
fffff880`15924878  00000000`00000000
fffff880`15924880  00000000`00000000
fffff880`15924888  fffff880`035cd8e0 BasicDisplay!CopyBitsTo_4+0x3d0
fffff880`15924890  00000000`00000000
fffff880`15924898  00000000`00000000
fffff880`159248a0  00000000`00000000
fffff880`159248a8  00000000`00000000
fffff880`159248b0  00000004`00008007
fffff880`159248b8  00000001`00000018
fffff880`159248c0  00000018`00000001
fffff880`159248c8  fffff880`159249f0
fffff880`159248d0  fffff880`15924a18
fffff880`159248d8  fffff880`00bdc0a8
fffff880`159248e0  00000000`00000001
fffff880`159248e8  00000000`00000031
fffff880`159248f0  00000000`00000000
fffff880`159248f8  00000000`00000000
fffff880`15924900  00000000`00000000
fffff880`15924908  00000000`00000000
fffff880`15924910  fffff880`00000000
fffff880`15924918  00000000`00000000
fffff880`15924920  00000000`00000000
fffff880`15924928  fffff880`035cd8e0 BasicDisplay!CopyBitsTo_4+0x3d0
```

```
fffff880`15924930  00000280`00000000
fffff880`15924938  0000003c`000001e0
fffff880`15924940  00000000`00000000
fffff880`15924948  00000000`00000000
fffff880`15924950  ffff3753`3e069d3e
fffff880`15924958  00000001`0000000d
fffff880`15924960  00000000`00000018
fffff880`15924968  00000000`00000004
fffff880`15924970  00000000`00000000
fffff880`15924978  fffff880`00bdc078
fffff880`15924980  00000000`ffffff23
fffff880`15924988  fffff880`035cdd4e BasicDisplay!BltBits+0x42
fffff880`15924990  00000000`fffffe73
fffff880`15924998  fffff880`15924a51
fffff880`159249a0  fffffa80`02c55d20
fffff880`159249a8  00000000`0000000d
fffff880`159249b0  00000004`0000e001
fffff880`159249b8  00000001`00000018
fffff880`159249c0  00000000`0000018d
fffff880`159249c8  fffff880`035cd416 BasicDisplay!BddDdiSystemDisplayWrite+0x11e
fffff880`159249d0  fffff880`15924a18
fffff880`159249d8  fffff880`159249f0
fffff880`159249e0  00000000`00000001
fffff880`159249e8  00000000`00000030
fffff880`159249f0  fffff880`00bdc078
fffff880`159249f8  00000004`00000002
fffff880`15924a00  ffffff23`fffffe73
fffff880`15924a08  00000004`00000001
fffff880`15924a10  00000000`00000018
fffff880`15924a18  fffff880`03c6b000
fffff880`15924a20  00000004`00000050
fffff880`15924a28  00000000`00000000
fffff880`15924a30  00000280`00000001
fffff880`15924a38  fffff880`000001e0
fffff880`15924a40  000000dd`0000018d
fffff880`15924a48  000000f5`00000191
fffff880`15924a50  ffff3753`3e069c7e
fffff880`15924a58  00000000`00000000
fffff880`15924a60  00000000`00000004
fffff880`15924a68  00000000`00000001
fffff880`15924a70  00000000`00000018
fffff880`15924a78  00000000`00000018
fffff880`15924a80  00000000`00000004
fffff880`15924a88  00000000`00000004
fffff880`15924a90  fffff880`15924b00
fffff880`15924a98  fffff880`03418c9e dxgkrnl!DpiSystemDisplayWrite+0xee
fffff880`15924aa0  fffff880`00bdc0a7
fffff880`15924aa8  00000000`00000000
fffff880`15924ab0  00000000`00000001
fffff880`15924ab8  fffff802`b3bc7f84 nt!RaspAntiAlias+0x104
fffff880`15924ac0  fffff880`00000002
fffff880`15924ac8  fffff880`0000018d
fffff880`15924ad0  00000000`000000dd
fffff880`15924ad8  00000000`00000001
fffff880`15924ae0  00000000`00000000
fffff880`15924ae8  00000000`00000018
fffff880`15924af0  fffff880`15924ce8
fffff880`15924af8  fffff880`15924b99
fffff880`15924b00  fffff880`15924b99
fffff880`15924b08  fffff802`b3bd77f6 nt!GxpWriteFrameBufferPixels+0x13e
```

```
fffff880`15924b10  fffff880`00bdc030
fffff880`15924b18  fffff880`15924ce8
fffff880`15924b20  fffff880`15924b60
fffff880`15924b28  fffff802`b3bc7e02 nt!BgpRasPrintGlyph+0x28a
fffff880`15924b30  fffff880`15924b60
fffff880`15924b38  00000004`00000018
fffff880`15924b40  00014af4`00000001
fffff880`15924b48  fffff880`00bdc078
fffff880`15924b50  fffffa80`00000004
fffff880`15924b58  fffff880`00000000
fffff880`15924b60  00000004`00000018
fffff880`15924b68  00014af4`00000004
fffff880`15924b70  fffffa80`00000000
fffff880`15924b78  fffff880`00bdc078
fffff880`15924b80  00000000`00000001
fffff880`15924b88  fffff880`04440f79 dump_dumpata!IdeDumpNotification+0x1e1
fffff880`15924b90  00000000`00000000
fffff880`15924b98  00000000`00000002
fffff880`15924ba0  fffff880`15924c20
fffff880`15924ba8  fffff880`15924ce0
fffff880`15924bb0  fffffa80`03337000
fffff880`15924bb8  fffff880`04440f39 dump_dumpata!IdeDumpNotification+0x1a1
fffff880`15924bc0  00000000`00000200
fffff880`15924bc8  fffff802`b3d17fe0 nt!BcpCharacterCache
fffff880`15924bd0  00000000`00000000
fffff880`15924bd8  fffffa80`018289a0
fffff880`15924be0  fffff880`15924ce8
fffff880`15924be8  fffff880`00bdc030
fffff880`15924bf0  00000000`00000001
fffff880`15924bf8  fffff880`04442614 dump_dumpata!AtaPortGetPhysicalAddress+0x2c
fffff880`15924c00  00000000`000050e0
fffff880`15924c08  fffffa80`03337260
fffff880`15924c10  00000000`00000000
fffff880`15924c18  fffff880`00baebc9
fffff880`15924c20  fffffa80`0000000c
fffff880`15924c28  fffffa80`03337798
fffff880`15924c30  00000000`7afe7000
fffff880`15924c38  fffffa80`027e7000
fffff880`15924c40  00000000`00000000
fffff880`15924c48  fffff802`b3b15490 nt!RtlDecompressFragmentProcs
fffff880`15924c50  fffff880`00000000
fffff880`15924c58  fffffa80`03337798
fffff880`15924c60  fffffa80`033375a8
fffff880`15924c68  fffff880`0444e8ce*** ERROR: Module load completed but symbols could not be
loaded for dump_atapi.sys
 dump_atapi+0x28ce
fffff880`15924c70  00000000`00000000
fffff880`15924c78  fffff880`00bdc030
fffff880`15924c80  ffff7cad`450c35aa
fffff880`15924c88  fffff880`0444e7bc dump_atapi+0x27bc
fffff880`15924c90  00000000`00000103
fffff880`15924c98  fffffa80`033377a0
fffff880`15924ca0  00000000`00000000
fffff880`15924ca8  00000000`00000001
fffff880`15924cb0  fffffa80`03337798
fffff880`15924cb8  fffff880`0444e297 dump_atapi+0x2297
fffff880`15924cc0  fffffa80`033375a8
fffff880`15924cc8  fffffa80`033375f0
fffff880`15924cd0  fffffa80`033375f0
fffff880`15924cd8  fffffa80`03337798
```

```
fffff880`15924ce0  00000000`00000000
fffff880`15924ce8  fffff880`0444e0f4 dump_atapi+0x20f4
fffff880`15924cf0  fffffa80`033375f0
fffff880`15924cf8  fffff802`b3a24d07 hal!IoMapTransfer+0x1b
fffff880`15924d00  00000000`00000103
fffff880`15924d08  fffff802`b3a3b110 hal!HalpTimerStallExecutionProcessor+0x161
fffff880`15924d10  00000000`00000000
fffff880`15924d18  fffff880`15924ec8
fffff880`15924d20  fffffa80`03337798
fffff880`15924d28  fffff880`0444deb1 dump_atapi+0x1eb1
fffff880`15924d30  fffffa80`03337650
fffff880`15924d38  fffff880`0444d6c8 dump_atapi+0x16c8
fffff880`15924d40  fffff880`15924f00
fffff880`15924d48  00000000`00000000
fffff880`15924d50  fffff157`9399fa4b
fffff880`15924d58  fffff880`0444d678 dump_atapi+0x1678
fffff880`15924d60  00000000`00000103
fffff880`15924d68  fffffa80`033371c0
fffff880`15924d70  00000000`000003e8
fffff880`15924d78  fffff880`15924ec8
fffff880`15924d80  00000000`00000103
fffff880`15924d88  fffffa80`033371c0
fffff880`15924d90  00000000`000000e6
fffff880`15924d98  fffff880`04440cab dump_dumpata!IdeDumpPollInterrupt+0x37
fffff880`15924da0  00000000`00000000
fffff880`15924da8  00000000`00000000
fffff880`15924db0  00000000`ffffffff
fffff880`15924db8  00000000`ffffff44
fffff880`15924dc0  fffffa80`033371c0
fffff880`15924dc8  fffff880`04441982 dump_dumpata!IdeDumpWaitOnRequest+0xce
fffff880`15924dd0  fffffa80`03337001
fffff880`15924dd8  00000000`ffffffff
fffff880`15924de0  00000000`ffffffff
fffff880`15924de8  00000000`ffffffff
fffff880`15924df0  00000000`00000000
fffff880`15924df8  00000000`00000000
fffff880`15924e00  00000000`ffffffff
fffff880`15924e08  fffff880`04440794 dump_dumpata!IdeDumpIoIssue+0x110
fffff880`15924e10  fffffa80`03337000
fffff880`15924e18  fffffa80`03337000
fffff880`15924e20  fffff880`15924f00
fffff880`15924e28  00000000`00000000
fffff880`15924e30  fffffa80`033371c0
fffff880`15924e38  fffffa80`027b0103
fffff880`15924e40  00000000`00000020
fffff880`15924e48  00000000`00000002
fffff880`15924e50  00000000`00010000
fffff880`15924e58  fffff880`021e8097 crashdmp!CrashdmpWriteRoutine+0x4f
fffff880`15924e60  00000000`066e2000
fffff880`15924e68  fffff880`15924ec8
fffff880`15924e70  fffff880`15924f00
fffff880`15924e78  fffffa80`027b5950
fffff880`15924e80  00000000`13746000
fffff880`15924e88  fffff880`021ed3e0 crashdmp!Context+0x30
fffff880`15924e90  00000000`13746000
fffff880`15924e98  fffff880`021e62dc crashdmp!WritePageSpanToDisk+0x200
fffff880`15924ea0  00000000`066e2000
fffff880`15924ea8  fffff880`15924fa0
fffff880`15924eb0  fffff880`021ed3e0 crashdmp!Context+0x30
fffff880`15924eb8  fffff880`00000002
```

```
fffff880`15924ec0  fffff880`00000000
fffff880`15924ec8  0000000d`09886000
fffff880`15924ed0  fffff880`021e8048 crashdmp!CrashdmpWriteRoutine
fffff880`15924ed8  fffff880`021e812c crashdmp!CrashdmpWritePendingRoutine
fffff880`15924ee0  00000000`00010000
fffff880`15924ee8  00000000`0002dc63
fffff880`15924ef0  fffff880`021ed3e0 crashdmp!Context+0x30
fffff880`15924ef8  fffff802`b3b8149d nt!vsnprintf+0x11
fffff880`15924f00  00000000`00000000
fffff880`15924f08  00000000`20030000
fffff880`15924f10  00000000`00000000
fffff880`15924f18  fffff880`00841000
fffff880`15924f20  fffff880`00841000
fffff880`15924f28  00000000`00010000
fffff880`15924f30  00000000`0002dc63
fffff880`15924f38  00000000`0002dc64
fffff880`15924f40  00000000`0002dc65
fffff880`15924f48  00000000`0002dc66
fffff880`15924f50  00000000`0002dc67
fffff880`15924f58  00000000`0002dc68
fffff880`15924f60  00000000`0002dc69
fffff880`15924f68  00000000`0002dc6a
fffff880`15924f70  00000000`0002dc6b
fffff880`15924f78  00000000`0002dc6c
fffff880`15924f80  00000000`0002dc6d
fffff880`15924f88  00000000`0002dc6e
fffff880`15924f90  00000000`0002dc6f
fffff880`15924f98  00000000`0002dc70
fffff880`15924fa0  00000000`0002dc71
fffff880`15924fa8  00000000`0002dc72
fffff880`15924fb0  00000000`00000000
fffff880`15924fb8  00000000`0017c85d
fffff880`15924fc0  ffffcbba`a93076e8
fffff880`15924fc8  fffff802`b3d17590 nt!NtVhdBootFile+0x15d8
fffff880`15924fd0  fffff880`15925510
fffff880`15924fd8  00000000`0004fae9
fffff880`15924fe0  00000000`00000000
fffff880`15924fe8  00000000`0002dc63
fffff880`15924ff0  00000000`00000000
fffff880`15924ff8  00000000`00000010
fffff880`15925000  fffff880`15925400
fffff880`15925008  fffff880`021e5e2a crashdmp!WriteBitmapDump+0x25e
fffff880`15925010  fffff880`159250d0
fffff880`15925018  fffff880`021ed3e0 crashdmp!Context+0x30
fffff880`15925020  00000000`00000050
fffff880`15925028  00000000`00000000
fffff880`15925030  fffff880`00000050
fffff880`15925038  fffff880`00000001
fffff880`15925040  00000000`00066bec
fffff880`15925048  fffff880`00016ae9
fffff880`15925050  00000000`00000000
fffff880`15925058  00000000`00000000
fffff880`15925060  00000000`00000000
fffff880`15925068  00000000`00066c63
fffff880`15925070  00000000`0007d74c
fffff880`15925078  fffffa80`02c02038
fffff880`15925080  00000000`00000010
fffff880`15925088  fffff802`b3bfe96c nt!KiBugCheckProgress
fffff880`15925090  00000000`0007d6d5
fffff880`15925098  00000000`00066bec
```

```
fffff880`159250a0  fffff880`021ed3e0 crashdmp!Context+0x30
fffff880`159250a8  fffff802`b3bfe96c nt!KiBugCheckProgress
fffff880`159250b0  00000000`00000000
fffff880`159250b8  fffffa80`02c02030
fffff880`159250c0  00000000`0007d6d5
fffff880`159250c8  00000000`00000000
fffff880`159250d0  20676e69`706d7544
fffff880`159250d8  6c616369`73796870
fffff880`159250e0  2079726f`6d656d20
fffff880`159250e8  3a6b7369`64206f74
fffff880`159250f0  000d2025`30382020
fffff880`159250f8  00000000`00000000
fffff880`15925100  00000000`00000000
fffff880`15925108  00000000`00000000
fffff880`15925110  00000000`00000000
fffff880`15925118  00000000`00000000
fffff880`15925120  00000000`00000000
fffff880`15925128  00000000`00000000
fffff880`15925130  ffffcbba`00000000
fffff880`15925138  00000000`0badf00d
fffff880`15925140  ffffcbba`a9306858
fffff880`15925148  fffff802`b3bfe96c nt!KiBugCheckProgress
fffff880`15925150  fffff802`b3bfe96c nt!KiBugCheckProgress
fffff880`15925158  fffff802`b3bfe96c nt!KiBugCheckProgress
fffff880`15925160  00000000`00000001
fffff880`15925168  00000000`00000000
fffff880`15925170  00000000`0000f08b
fffff880`15925178  fffff880`021e5985 crashdmp!DumpWrite+0x1c5
fffff880`15925180  fffff880`021ed3e0 crashdmp!Context+0x30
fffff880`15925188  fffff880`021ed3e0 crashdmp!Context+0x30
fffff880`15925190  fffff880`021ed3e0 crashdmp!Context+0x30
fffff880`15925198  fffff802`b3d7f100 nt!KiInitialPCR+0x100
fffff880`159251a0  fffff802`b3bfe96c nt!KiBugCheckProgress
fffff880`159251a8  00000000`00000001
fffff880`159251b0  fffff802`b3d7f100 nt!KiInitialPCR+0x100
fffff880`159251b8  fffff880`021e4a4e crashdmp!CrashdmpWrite+0x9e
fffff880`159251c0  00000000`00000000
fffff880`159251c8  fffff880`15925490
fffff880`159251d0  fffff802`b3d60200 nt!IopTriageDumpDataBlocks
fffff880`159251d8  fffff802`b3bfe96c nt!KiBugCheckProgress
fffff880`159251e0  00000000`00000001
fffff880`159251e8  fffff802`b3bf4ea7 nt!IoWriteCrashDump+0x5e3
fffff880`159251f0  00000000`00000000
fffff880`159251f8  fffff880`15925490
fffff880`15925200  fffff802`b3d5ae00 nt!KeBugCheckAddPagesCallbackListHead
fffff880`15925208  00000000`00000001
fffff880`15925210  00300030`00300030
fffff880`15925218  00300030`00300030
fffff880`15925220  00300030`00300030
fffff880`15925228  00300030`00300030
fffff880`15925230  00300078`00300000
fffff880`15925238  00300030`00300030
fffff880`15925240  00300030`00300030
fffff880`15925248  00300030`00300030
fffff880`15925250  000000ef`00300130
fffff880`15925258  00000000`00000000
fffff880`15925260  00000000`00000000
fffff880`15925268  fffff802`b3d5ae00 nt!KeBugCheckAddPagesCallbackListHead
fffff880`15925270  fffffa80`02e6b1c0
fffff880`15925278  fffff802`b3d5ae00 nt!KeBugCheckAddPagesCallbackListHead
```

```
fffff880`15925280  fffffa80`02c02000
fffff880`15925288  0000000a`000d0044
fffff880`15925290  00000000`00000000
fffff880`15925298  00000000`000000ef
fffff880`159252a0  00000000`00000000
fffff880`159252a8  00000000`00000000
fffff880`159252b0  fffff880`15925510
fffff880`159252b8  00000000`00000000
fffff880`159252c0  00000000`00000000
fffff880`159252c8  00000000`00000000
fffff880`159252d0  00000000`00000000
fffff880`159252d8  fffff802`b3bfe96c nt!KiBugCheckProgress
fffff880`159252e0  fffffa80`02c02000
fffff880`159252e8  fffff802`b3bf4710 nt!IoSetDumpRange
fffff880`159252f0  fffff802`b3bf4670 nt!IoFreeDumpRange
fffff880`159252f8  fffffa80`02e6b1c0
fffff880`15925300  00000000`00000000
fffff880`15925308  00000000`00000000
fffff880`15925310  00000000`0007d74c
fffff880`15925318  fffffa80`02c02038
fffff880`15925320  fffffa80`02e6b1c0
fffff880`15925328  00000000`00000000
fffff880`15925330  00000000`00000000
fffff880`15925338  00000000`00000000
fffff880`15925340  ffff7cad`450c285a
fffff880`15925348  fffff802`b3d7f180 nt!KiInitialPCR+0x180
fffff880`15925350  00000000`00000000
fffff880`15925358  fffff802`b3d7f180 nt!KiInitialPCR+0x180
fffff880`15925360  00000000`00000000
fffff880`15925368  00000000`000000ef
fffff880`15925370  fffffa80`02e6b100
fffff880`15925378  00000000`00000001
fffff880`15925380  00000000`00000000
fffff880`15925388  fffff802`b3bfe5b0 nt!KeBugCheck2+0x9c1
fffff880`15925390  fffff802`b3d1a5a0 nt!EtwpBugCheckCallback
fffff880`15925398  fffff802`b3d5adf0 nt!KeBugCheckReasonCallbackListHead
fffff880`159253a0  fffff802`b3d5adf0 nt!KeBugCheckReasonCallbackListHead
fffff880`159253a8  00000000`00000001
fffff880`159253b0  00000000`00000000
fffff880`159253b8  fffff880`15925510
fffff880`159253c0  fffffa80`03db4740
fffff880`159253c8  fffff802`b3bfe96c nt!KiBugCheckProgress
fffff880`159253d0  fffff8a0`02c8dc01
fffff880`159253d8  fffff802`b3f5bbf4 nt!CmpCallCallBacks+0x3e4
fffff880`159253e0  01010001`0101dc40
fffff880`159253e8  00000000`00000000
fffff880`159253f0  fffff880`159255d0
fffff880`159253f8  00000000`00000000
fffff880`15925400  00000000`00000000
fffff880`15925408  fffff802`b3b3c95d nt!ExQueueWorkItem+0x1fd
fffff880`15925410  fffff8a0`00000000
fffff880`15925418  fffff802`b3d7f180 nt!KiInitialPCR+0x180
fffff880`15925420  fffffa80`03db4740
fffff880`15925428  fffff800`00000000
fffff880`15925430  ffffffff`ffffffff
fffff880`15925438  fffff802`b3bfe96c nt!KiBugCheckProgress
fffff880`15925440  fffff8a0`013d2f0c
fffff880`15925448  fffff802`b3d0d000 nt!ExNode0
fffff880`15925450  fffff880`15925b10
fffff880`15925458  00000000`0fa79f0a
```

```
fffff880`15925460  00000000`00140001
fffff880`15925468  00000000`00000002
fffff880`15925470  fffff880`15925500
fffff880`15925478  ffffffff`ffffffff
fffff880`15925480  fffff880`15925b10
fffff880`15925488  00000000`c0000034
fffff880`15925490  00000000`00000000
fffff880`15925498  00000000`00000001
fffff880`159254a0  fffffa80`03de4750
fffff880`159254a8  fffff8a0`00935380
fffff880`159254b0  00000000`00000000
fffff880`159254b8  fffff802`b3ebca64 nt!CmpParseKey+0x865
fffff880`159254c0  fffff880`0000001d
fffff880`159254c8  fffff880`15925698
fffff880`159254d0  fffff8a0`00b49000
fffff880`159254d8  fffff880`0000001d
fffff880`159254e0  00000000`00000000
fffff880`159254e8  fffff880`15925628
fffff880`159254f0  fffff880`159255d8
fffff880`159254f8  fffff880`15925580
fffff880`15925500  fffff880`15925b10
fffff880`15925508  fffff880`15925620
fffff880`15925510  00000000`00000000
fffff880`15925518  00000000`00000000
fffff880`15925520  00000000`00000000
fffff880`15925528  00000000`00000000
fffff880`15925530  00000000`00000000
fffff880`15925538  00000000`00000000
fffff880`15925540  00001f80`0010000f
fffff880`15925548  0053002b`002b0010
fffff880`15925550  00000246`0018002b
fffff880`15925558  00000000`00000000
fffff880`15925560  00000000`00000000
fffff880`15925568  00000000`00000000
fffff880`15925570  00000000`00000000
fffff880`15925578  00000000`00000000
fffff880`15925580  00000000`00000000
fffff880`15925588  fffff880`15925b03
fffff880`15925590  00000000`000000ef
fffff880`15925598  fffffa80`02e6b1c0
fffff880`159255a0  fffffa80`02e6b100
fffff880`159255a8  fffff880`15925ae8
fffff880`159255b0  00000000`00000001
fffff880`159255b8  00000000`00000000
fffff880`159255c0  fffffa80`02e6b1c0
fffff880`159255c8  00000000`00000000
fffff880`159255d0  00000000`00000000
fffff880`159255d8  00000000`144d2c09
fffff880`159255e0  fffff880`15925c38
fffff880`159255e8  00000000`00000001
fffff880`159255f0  00000000`00000000
fffff880`159255f8  fffffa80`03db4740
fffff880`15925600  fffffa80`03db4740
fffff880`15925608  fffff802`b3b03d40 nt!KeBugCheckEx
fffff880`15925610  00000000`0000027f
fffff880`15925618  00000000`00000000
fffff880`15925620  00000000`00000000
fffff880`15925628  00000000`00001f80
fffff880`15925630  00000000`00000000
fffff880`15925638  00000000`00000000
```

```
fffff880`15925640  00000000`00000000
[...]
fffff880`159259d8  00000000`00000000
fffff880`159259e0  fffffa80`02e6b100
fffff880`159259e8  00000000`00000000
fffff880`159259f0  00000000`ffffffff
fffff880`159259f8  00000000`00f800ca
fffff880`15925a00  fffff8a0`005e7560
fffff880`15925a08  fffff802`b3d7f180 nt!KiInitialPCR+0x180
fffff880`15925a10  00000000`00000001
fffff880`15925a18  000000f0`6e86e760
fffff880`15925a20  00000000`00000001
fffff880`15925a28  00000000`00000000
fffff880`15925a30  00000000`00000000
fffff880`15925a38  00000000`00000000
fffff880`15925a40  fffff880`15925cc0
fffff880`15925a48  fffff802`b3ec1e8d nt!CmOpenKey+0x31c
fffff880`15925a50  00000000`00000000
fffff880`15925a58  000000f0`6e86e780
fffff880`15925a60  00000000`00000001
fffff880`15925a68  fffffa80`03db4740
fffff880`15925a70  fffffa80`03db4740
fffff880`15925a78  00000000`00000000
fffff880`15925a80  00000000`00000001
fffff880`15925a88  fffffa80`02e6b1c0
fffff880`15925a90  00000000`00000000
fffff880`15925a98  fffffa80`02e6b100
fffff880`15925aa0  00000000`00000001
fffff880`15925aa8  fffff802`b3b03e44 nt!KeBugCheckEx+0x104
fffff880`15925ab0  00000000`00000000
fffff880`15925ab8  00000000`00000000
fffff880`15925ac0  00000000`00000000
fffff880`15925ac8  00000000`00000001
fffff880`15925ad0  00000000`00000000
fffff880`15925ad8  00000000`00000000
fffff880`15925ae0  00000000`00000246
fffff880`15925ae8  fffff802`b400f0dd nt!PspCatchCriticalBreak+0xad
fffff880`15925af0  00000000`000000ef
fffff880`15925af8  fffffa80`02e6b1c0
fffff880`15925b00  00000000`00000000
fffff880`15925b08  00000000`00000000
fffff880`15925b10  00000000`00000000
fffff880`15925b18  00000000`00000000
fffff880`15925b20  fffffa80`02e6b1c0
fffff880`15925b28  fffff802`b3ea8f6d nt! ?? ::NNGAKEGL::`string'+0x46f60
fffff880`15925b30  fffffa80`02e6b1c0
fffff880`15925b38  00000000`144d2c01
fffff880`15925b40  00000000`00000000
fffff880`15925b48  ffff7cad`450c235a
fffff880`15925b50  fffffa80`03db4740
fffff880`15925b58  00000000`00000001
fffff880`15925b60  00000000`00000000
fffff880`15925b68  00000000`00000000
fffff880`15925b70  00000000`00000000
fffff880`15925b78  00000000`00000000
fffff880`15925b80  00000000`144d2c01
fffff880`15925b88  fffff802`b3ea8019 nt!PspTerminateProcess+0x6d
fffff880`15925b90  fffffa80`02e6b1c0
fffff880`15925b98  00000000`144d2c01
fffff880`15925ba0  fffffa80`02e6b1c0
```

```
fffff880`15925ba8  00000000`00000000
fffff880`15925bb0  00000000`00000001
fffff880`15925bb8  00000000`00000601
fffff880`15925bc0  fffffa80`03db4740
fffff880`15925bc8  fffff802`b3ea7e52 nt!NtTerminateProcess+0x9e
fffff880`15925bd0  ffffffff`ffffffff
fffff880`15925bd8  fffffa80`02d74180
fffff880`15925be0  fffffa80`02e6b1c0
fffff880`15925be8  00000000`00000001
fffff880`15925bf0  fffffa80`65547350
fffff880`15925bf8  fffff880`15925c40
fffff880`15925c00  00000000`00000000
fffff880`15925c08  ffff7cad`450c223a
fffff880`15925c10  000000f0`6edd7480
fffff880`15925c18  00000000`00000648
fffff880`15925c20  00000000`00000190
fffff880`15925c28  00000000`00000000
fffff880`15925c30  00000000`00000000
fffff880`15925c38  fffff802`b3b02d53 nt!KiSystemServiceCopyEnd+0x13
fffff880`15925c40  fffffa80`02e6b1c0
fffff880`15925c48  fffffa80`03db4740
fffff880`15925c50  fffff880`15925cc0
fffff880`15925c58  00000000`00000000
fffff880`15925c60  000000f0`00000000
fffff880`15925c68  00001fa0`02080000
fffff880`15925c70  00000000`00000000
fffff880`15925c78  00000000`000006b4
fffff880`15925c80  000007fe`f2956890 COMCTL32!DirectUI::InvokeHelper::s_uInvokeHelperMsg+0x88
fffff880`15925c88  000000f0`6e86f068
fffff880`15925c90  00000000`00000000
fffff880`15925c98  00000000`00000000
fffff880`15925ca0  00000000`00000246
fffff880`15925ca8  000007f7`70b7d000
fffff880`15925cb0  00000000`00000000
fffff880`15925cb8  00000000`00000000
fffff880`15925cc0  00000000`00000000
fffff880`15925cc8  00000000`00000000
fffff880`15925cd0  00000000`00000000
fffff880`15925cd8  00000000`00000000
fffff880`15925ce0  00000000`00000000
fffff880`15925ce8  00000000`00000000
fffff880`15925cf0  00000000`00000000
fffff880`15925cf8  00000000`00000000
fffff880`15925d00  00000000`00000000
fffff880`15925d08  00000000`00000000
fffff880`15925d10  000007fe`f2901000
COMCTL32!DirectUI::StyleSheetCache::CCacheThread::Initialize+0x54
fffff880`15925d18  00000000`00000000
fffff880`15925d20  00000000`00000000
fffff880`15925d28  00000000`00000000
fffff880`15925d30  00000000`00000000
fffff880`15925d38  00000000`00000000
fffff880`15925d40  00000000`00000000
fffff880`15925d48  00000000`00000000
fffff880`15925d50  00000000`00000000
fffff880`15925d58  00000000`00000000
fffff880`15925d60  00000000`00000000
fffff880`15925d68  00000000`00000000
fffff880`15925d70  00000000`00000000
fffff880`15925d78  00000000`00000000
```

```
fffff880`15925d80  00000000`00000648
fffff880`15925d88  00000000`00000001
fffff880`15925d90  00000000`00000000
fffff880`15925d98  000000f0`6e86f470
fffff880`15925da0  00000000`00000014
fffff880`15925da8  000007fe`f7ec2eaa ntdll!NtTerminateProcess+0xa
fffff880`15925db0  00000000`00000033
fffff880`15925db8  00000000`00000202
fffff880`15925dc0  000000f0`6e86f3e8
fffff880`15925dc8  00000000`0000002b
fffff880`15925dd0  fffff880`15926000
[...]
fffff880`15925ff0  00000000`00000000
fffff880`15925ff8  00000000`00000000
fffff880`15926000  ????????`????????
```

We can examine any suspicious module using **lmv** and **!lmi** commands.

```
0: kd> lmv m igdkmd64
start             end                 module name
fffff880`03e17000 fffff880`043fee00   igdkmd64   (pdb symbols)
C:\WinDbg.Docker.AWMA\mss\igdkmd64.pdb\32FCA049C8194A398B9BE29BAF0CA69C1\igdkmd64.pdb
    Loaded symbol image file: igdkmd64.sys
    Image path: \SystemRoot\system32\DRIVERS\igdkmd64.sys
    Image name: igdkmd64.sys
    Timestamp:        Fri Mar 23 04:33:47 2012 (4F6BFD2B)
    CheckSum:         005EBF0F
    ImageSize:        005E7E00
    Translations:     0000.04b0 0000.04e4 0409.04b0 0409.04e4

0: kd> !lmi igdkmd64
Loaded Module Info: [igdkmd64]
         Module: igdkmd64
   Base Address: fffff88003e17000
     Image Name: igdkmd64.sys
   Machine Type: 34404 (X64)
     Time Stamp: 4f6bfd2b Fri Mar 23 04:33:47 2012
           Size: 5e7e00
       CheckSum: 5ebf0f
Characteristics: 2022
Debug Data Dirs: Type  Size     VA  Pointer
             CODEVIEW    89, 4cf978,  4cf978 RSDS - GUID: {32FCA049-C819-4A39-8B9B-
E29BAF0CA69C}
               Age: 1, Pdb: D:\ccViews\autobuild1_BR-1203-
0FZG_15.12.75_Snapshot\gfx_Development\dump64\igfx\lh\release\AIM3Lib\igdkmd64.pdb
     Image Type: MEMORY   - Image read successfully from loaded memory.
    Symbol Type: PDB      - Symbols loaded successfully from symbol server.
        C:\WinDbg.Docker.AWMA\mss\igdkmd64.pdb\32FCA049C8194A398B9BE29BAF0CA69C1\igdkmd64.pdb
    Load Report: public symbols , not source indexed
        C:\WinDbg.Docker.AWMA\mss\igdkmd64.pdb\32FCA049C8194A398B9BE29BAF0CA69C1\igdkmd64.pdb
```

Note that this module has symbols that come from Microsoft symbol server so it should be Microsoft module. Additionally we can also inspect module header using **!dh** command. Now we search for strings using various commands like we did in user space:

```
0: kd> s-sa fffff88015920000 fffff88015926000
fffff880`1592341c  " CR"
fffff880`15923474  " CR"
fffff880`159235bc  " CR"
```

```
fffff880`1592388c  " CR"
fffff880`15923aa4  " CR"
fffff880`15923b94  " CR"
fffff880`15923cfc  " CR"
fffff880`15923db4  " CR"
fffff880`15923f84  " CR"
fffff880`159240dc  " CR"
fffff880`15924124  " CR"
fffff880`15924344  " CR"
fffff880`159243d4  " CR"
fffff880`15924424  " CR"
fffff880`15924953  ">S7"
fffff880`15924a53  ">S7"
fffff880`15924c08  "`r3"
fffff880`15924d30  "Pv3"
fffff880`15924e78  "PY{"
fffff880`159250d0  "Dumping physical memory to disk:"
fffff880`159250f0  "  80% "
fffff880`15925140  "Xh0"
fffff880`15925a00  "`u^"
fffff880`15925bf0  "PsTe"
```

```
0: kd> dpa fffff88015920000 fffff88015926000
[...]
fffff880`15925000  fffff880`15925400 ""
fffff880`15925008  fffff880`021e5e2a "D...D$4..$."
fffff880`15925010  fffff880`159250d0 "Dumping physical memory to disk:  80% ."
fffff880`15925018  fffff880`021ed3e0 "PY{......."
fffff880`15925020  00000000`00000050
[...]
```

Note that the stack page was saved to a dump file when the progress bar was at 80%.

12. Now we can list all processes and their stack traces. The first **!process** command type only lists the sort summary:

```
0: kd> !process 0 0
**** NT ACTIVE PROCESS DUMP ****
PROCESS fffffa800182e480
    SessionId: none  Cid: 0004    Peb: 00000000  ParentCid: 0000
    DirBase: 00187000  ObjectTable: fffff8a000003000  HandleCount: <Data Not Accessible>
    Image: System

PROCESS fffffa8002d78500
    SessionId: none  Cid: 011c    Peb: 7f6a68af000  ParentCid: 0004
    DirBase: 06696000  ObjectTable: fffff8a000b3b840  HandleCount: <Data Not Accessible>
    Image: smss.exe

PROCESS fffffa8002e6b1c0
    SessionId: 0  Cid: 0190    Peb: 7f7688e8000  ParentCid: 0188
    DirBase: 114d5000  ObjectTable: fffff8a001c6c680  HandleCount: <Data Not Accessible>
    Image: csrss.exe

PROCESS fffffa8002e7b940
    SessionId: 0  Cid: 01c4    Peb: 7f6f01fc000  ParentCid: 0188
    DirBase: 2449b000  ObjectTable: fffff8a00156ed80  HandleCount: <Data Not Accessible>
    Image: wininit.exe

PROCESS fffffa80033c3080
```

```
    SessionId: 0  Cid: 0220    Peb: 7f75ab5d000  ParentCid: 01c4
    DirBase: 2e23b000  ObjectTable: fffff8a0016a32c0  HandleCount: <Data Not Accessible>
    Image: services.exe

[...]
```

To list all thread stacks in detail, you can use the same command with different flags (**3f** is necessary to get the correct user space portion of stack traces for complete memory dumps):

```
0: kd> !process 0 3f
**** NT ACTIVE PROCESS DUMP ****
[...]
```

Note that we skip the output here because it fills a book.

Finally the last command show zombie processes at the end:

```
0: kd> !vm
Page File: \??\C:\pagefile.sys
  Current:   2359296 Kb  Free Space:   2272648 Kb
  Minimum:   2359296 Kb  Maximum:      6291456 Kb
Page File: \??\C:\swapfile.sys
  Current:    262144 Kb  Free Space:    262136 Kb
  Minimum:    262144 Kb  Maximum:      3082492 Kb

Physical Memory:           513749 (    2054996 Kb)
Available Pages:           216378 (     865512 Kb)
ResAvail Pages:            445904 (    1783616 Kb)
Locked IO Pages:                0 (          0 Kb)
Free System PTEs:        33460094 (  133840376 Kb)
Modified Pages:              5403 (      21612 Kb)
Modified PF Pages:           5400 (      21600 Kb)
Modified No Write Pages:        1 (          4 Kb)
NonPagedPool Usage:           784 (       3136 Kb)
NonPagedPoolNx Usage:        7868 (      31472 Kb)
NonPagedPool Max:          979551 (    3918204 Kb)
PagedPool  0:               17859 (      71436 Kb)
PagedPool  1:                3094 (      12376 Kb)
PagedPool  2:                1385 (       5540 Kb)
PagedPool  3:                1362 (       5448 Kb)
PagedPool  4:                1430 (       5720 Kb)
PagedPool Usage:            25130 (     100520 Kb)
PagedPool Maximum:      100663296 (  402653184 Kb)
Processor Commit:             510 (       2040 Kb)
Session Commit:              6322 (      25288 Kb)
Syspart SharedCommit 0
Shared Commit:              57010 (     228040 Kb)
Special Pool:                   0 (          0 Kb)
Kernel Stacks:               4259 (      17036 Kb)
Pages For MDLs:             16710 (      66840 Kb)
Pages For AWE:                  0 (          0 Kb)
NonPagedPool Commit:            0 (          0 Kb)
PagedPool Commit:           25146 (     100584 Kb)
Driver Commit:              10957 (      43828 Kb)
Boot Commit:                    0 (          0 Kb)
System PageTables:              0 (          0 Kb)
VAD/PageTable Bitmaps:       2013 (       8052 Kb)
```

```
ProcessLockedFilePages:         0 (          0 Kb)
Pagefile Hash Pages:            0 (          0 Kb)
Sum System Commit:         122927 (     491708 Kb)
Total Private:             124369 (     497476 Kb)
Misc/Transient Commit:      20092 (      80368 Kb)
Committed pages:           267388 (    1069552 Kb)
Commit limit:             1103573 (    4414292 Kb)

  Pid ImageName                        Commit   SharedCommit        Debt

  598 MsMpEng.exe                    68456 Kb           0 Kb        0 Kb
  6f8 dwm.exe                        52808 Kb           0 Kb        0 Kb
  3f0 svchost.exe                    50796 Kb           0 Kb        0 Kb
  d04 iexplore.exe                   36968 Kb           0 Kb        0 Kb
  314 svchost.exe                    35772 Kb           0 Kb        0 Kb
  d68 explorer.exe                   35596 Kb           0 Kb        0 Kb
  478 WWAHost.exe                    22296 Kb           0 Kb        0 Kb
  4e4 svchost.exe                    17124 Kb           0 Kb        0 Kb
  2f0 svchost.exe                    16204 Kb           0 Kb        0 Kb
  270 SearchIndexer.exe              15712 Kb           0 Kb        0 Kb
  f98 msiexec.exe                    14900 Kb           0 Kb        0 Kb
  bdc LiveComm.exe                   12740 Kb           0 Kb        0 Kb
  ca0 Taskmgr.exe                    11620 Kb           0 Kb        0 Kb
  3b8 svchost.exe                     9412 Kb           0 Kb        0 Kb
  c80 iexplore.exe                    9256 Kb           0 Kb        0 Kb
  a50 mspaint.exe                     8580 Kb           0 Kb        0 Kb
  360 svchost.exe                     7972 Kb           0 Kb        0 Kb
  2a0 taskhostex.exe                  7304 Kb           0 Kb        0 Kb
  8a8 svchost.exe                     6224 Kb           0 Kb        0 Kb
  7e8 svchost.exe                     6128 Kb           0 Kb        0 Kb
  ba8 wmpnetwk.exe                    5764 Kb           0 Kb        0 Kb
  228 lsass.exe                       4428 Kb           0 Kb        0 Kb
  4c8 spoolsv.exe                     4184 Kb           0 Kb        0 Kb
  220 services.exe                    4028 Kb           0 Kb        0 Kb
  3e4 RuntimeBroker.exe               3940 Kb           0 Kb        0 Kb
  2b0 svchost.exe                     3612 Kb           0 Kb        0 Kb
  63c dasHost.exe                     3524 Kb           0 Kb        0 Kb
  814 BackgroundTransferHost.e        3124 Kb           0 Kb        0 Kb
  288 svchost.exe                     2808 Kb           0 Kb        0 Kb
  e74 iexplore.exe                    2440 Kb           0 Kb        0 Kb
  dd0 browserchoice.exe               1980 Kb           0 Kb        0 Kb
  cdc csrss.exe                       1768 Kb           0 Kb        0 Kb
  e80 WmiPrvSE.exe                    1744 Kb           0 Kb        0 Kb
  2e4 svchost.exe                     1536 Kb           0 Kb        0 Kb
  bac dllhost.exe                     1444 Kb           0 Kb        0 Kb
  190 csrss.exe                       1396 Kb           0 Kb        0 Kb
  d7c notepad.exe                     1260 Kb           0 Kb        0 Kb
  a28 winlogon.exe                    1164 Kb           0 Kb        0 Kb
  1c4 wininit.exe                     1020 Kb           0 Kb        0 Kb
  11c smss.exe                         320 Kb           0 Kb        0 Kb
    4 System                           124 Kb           0 Kb        0 Kb
  dac LogonUI.exe                        0 Kb           0 Kb        0 Kb
  acc explorer.exe                       0 Kb           0 Kb        0 Kb
  a3c smss.exe                           0 Kb           0 Kb        0 Kb
```

13.	Now we check commands related to CPU consumption:

```
0: kd> !running -i

System Processors:  (0000000000000003)
  Idle Processors:  (0000000000000000)

       Prcbs             Current         (pri) Next            (pri) Idle
  0    fffff802b3d7f180  fffffa8003db4740 (13)                       fffff802b3dd9880  ................
  1    fffff880009e6180  fffffa80037b4080 (13)                       fffff880009f1dc0  ................
```

To quickly check the kernel space thread stack portion, we can use the **-t** flag:

```
0: kd> !running -i -t

System Processors:  (0000000000000003)
  Idle Processors:  (0000000000000000)

       Prcbs             Current         (pri) Next            (pri) Idle
  0    fffff802b3d7f180  fffffa8003db4740 (13)                       fffff802b3dd9880  ................

 # Child-SP          RetAddr           Call Site
00 fffff880`15925ae8 fffff802`b400f0dd     nt!KeBugCheckEx
01 fffff880`15925af0 fffff802`b3ea8f6d     nt!PspCatchCriticalBreak+0xad
02 fffff880`15925b30 fffff802`b3ea8019     nt! ?? ::NNGAKEGL::`string'+0x46f60
03 fffff880`15925b90 fffff802`b3ea7e52     nt!PspTerminateProcess+0x6d
04 fffff880`15925bd0 fffff802`b3b02d53     nt!NtTerminateProcess+0x9e
05 fffff880`15925c40 000007fe`f7ec2eaa     nt!KiSystemServiceCopyEnd+0x13
06 000000f0`6e86f3e8 00000000`00000000     ntdll!NtTerminateProcess+0xa

  1    fffff880009e6180  fffffa80037b4080 (13)                       fffff880009f1dc0  ................

 # Child-SP          RetAddr           Call Site
00 fffff880`159e39b0 fffff960`001862d3     win32k!xxxInternalDoPaint+0x19
01 fffff880`159e3a00 fffff960`001862d3     win32k!xxxInternalDoPaint+0x43
02 fffff880`159e3a50 fffff960`001862d3     win32k!xxxInternalDoPaint+0x43
03 fffff880`159e3aa0 fffff960`001862d3     win32k!xxxInternalDoPaint+0x43
04 fffff880`159e3af0 fffff960`001862d3     win32k!xxxInternalDoPaint+0x43
05 fffff880`159e3b40 fffff960`001862d3     win32k!xxxInternalDoPaint+0x43
06 fffff880`159e3b90 fffff960`0018608c     win32k!xxxInternalDoPaint+0x43
07 fffff880`159e3be0 fffff960`001532e3     win32k!xxxDoPaint+0x4c
08 fffff880`159e3c20 fffff960`00225974     win32k!xxxRealInternalGetMessage+0xa73
09 fffff880`159e3d40 fffff802`b3b02d53     win32k!NtUserRealInternalGetMessage+0x74
0a fffff880`159e3dd0 000007fe`f56c1b4a     nt!KiSystemServiceCopyEnd+0x13
0b 00000000`034af598 000007fe`f2a810fb     USER32!NtUserRealInternalGetMessage+0xa
0c 00000000`034af5a0 00000000`00000012     0x000007fe`f2a810fb
0d 00000000`034af5a8 000007fe`e5e31f20     0x12
0e 00000000`034af5b0 00000000`000100dc     0x000007fe`e5e31f20
0f 00000000`034af5b8 00000000`00000000     0x100dc
```

Unfortunately, it doesn’t show correct user space portion of the full stack trace so we use **!thread** command:

```
0: kd> !thread fffffa80037b4080 3f
THREAD fffffa80037b4080  Cid 0d68.0638  Teb: 000007f68f179000 Win32Thread: fffff9010063e5b0 RUNNING on processor 1
Not impersonating
DeviceMap                 fffff8a000290b20
Owning Process            fffffa8003ed3600       Image:         explorer.exe
Attached Process          N/A            Image:         N/A
Wait Start TickCount      15741128       Ticks: 0
Context Switch Count      18325          IdealProcessor: 1
UserTime                  00:00:00.280
KernelTime                00:00:00.405
Win32 Start Address SHCORE!COplockFileHandle::v_GetHandlerCLSID (0x000007fef2ef4020)
Stack Init fffff880159e3fd0 Current fffff880171fc7f0
Base fffff880159e4000 Limit fffff880159de000 Call 0000000000000000
Priority 13 BasePriority 9 PriorityDecrement 2 IoPriority 2 PagePriority 5
```

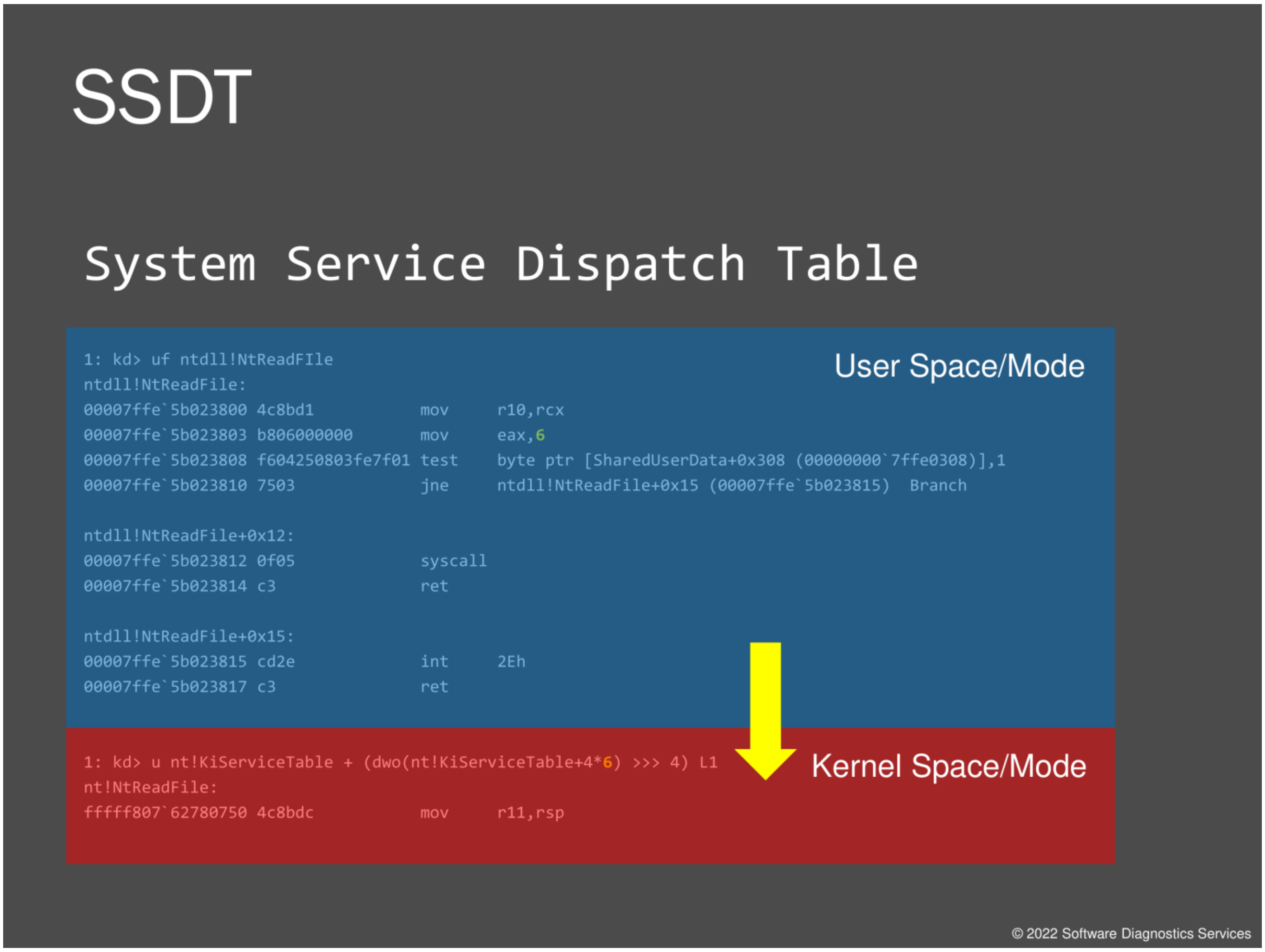
```
Child-SP          RetAddr           Call Site
fffff880`159e39b0 fffff960`001862d3 win32k!xxxInternalDoPaint+0x19
fffff880`159e3a00 fffff960`001862d3 win32k!xxxInternalDoPaint+0x43
fffff880`159e3a50 fffff960`001862d3 win32k!xxxInternalDoPaint+0x43
fffff880`159e3aa0 fffff960`001862d3 win32k!xxxInternalDoPaint+0x43
fffff880`159e3af0 fffff960`001862d3 win32k!xxxInternalDoPaint+0x43
fffff880`159e3b40 fffff960`001862d3 win32k!xxxInternalDoPaint+0x43
fffff880`159e3b90 fffff960`0018608c win32k!xxxInternalDoPaint+0x43
fffff880`159e3be0 fffff960`001532e3 win32k!xxxDoPaint+0x4c
fffff880`159e3c20 fffff960`00225974 win32k!xxxRealInternalGetMessage+0xa73
fffff880`159e3d40 fffff802`b3b02d53 win32k!NtUserRealInternalGetMessage+0x74
fffff880`159e3dd0 000007fe`f56c1b4a nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @ fffff880`159e3e40)
00000000`034af598 000007fe`f2a810fb USER32!NtUserRealInternalGetMessage+0xa
00000000`034af5a0 000007fe`f2a8120b DUser!CoreSC::xwProcessNL+0xe7
00000000`034af670 000007fe`f56c1bad DUser!MphProcessMessage+0xb3
00000000`034af6d0 000007fe`f7ec4b67 USER32!_ClientGetMessageMPH+0x3d
00000000`034af760 000007fe`f56c120a ntdll!KiUserCallbackDispatcherContinue (TrapFrame @ 00000000`034af628)
00000000`034af7d8 000007fe`f56c1250 USER32!NtUserPeekMessage+0xa
00000000`034af7e0 000007fe`f56c1145 USER32!PeekMessage+0x2c
00000000`034af820 000007f6`8f66105a USER32!PeekMessageW+0x85
00000000`034af860 000007f6`8f68b41e Explorer!CTray::_MessageLoop+0x4b
00000000`034af8f0 000007fe`f2ef410c Explorer!CTray::MainThreadProc+0x86
00000000`034af920 000007fe`f601167e SHCORE!COplockFileHandle::v_GetHandlerCLSID+0x12c
00000000`034afa10 000007fe`f7ee3501 KERNEL32!BaseThreadInitThunk+0x1a
00000000`034afa40 00000000`00000000 ntdll!RtlUserThreadStart+0x1d
```

And finally, for this exercise, we try the **!ready** command to list threads ready for execution:

```
0: kd> !ready
Processor 0: Ready Threads at priority 12
    THREAD fffffa80040667c0  Cid 0d68.0d3c  Teb: 000007f68f026000 Win32Thread: fffff90103f08b90 READY on processor 0
Processor 1: Ready Threads at priority 12
    THREAD fffffa8001da2380  Cid 0004.0f28  Teb: 0000000000000000 Win32Thread: 0000000000000000 READY on processor 1
Processor 1: Ready Threads at priority 10
    THREAD fffffa8003f0ca00  Cid 0d68.03b4  Teb: 000007f68f048000 Win32Thread: fffff90103ede780 READY on processor 1
    THREAD fffffa8002cdf300  Cid 0d68.0854  Teb: 000007f68f03c000 Win32Thread: fffff90103f544e0 READY on processor 1
```

14. Close the log file:

```
0: kd> .logclose
Closing open log file C:\AWMA-Dumps\M4.log
```



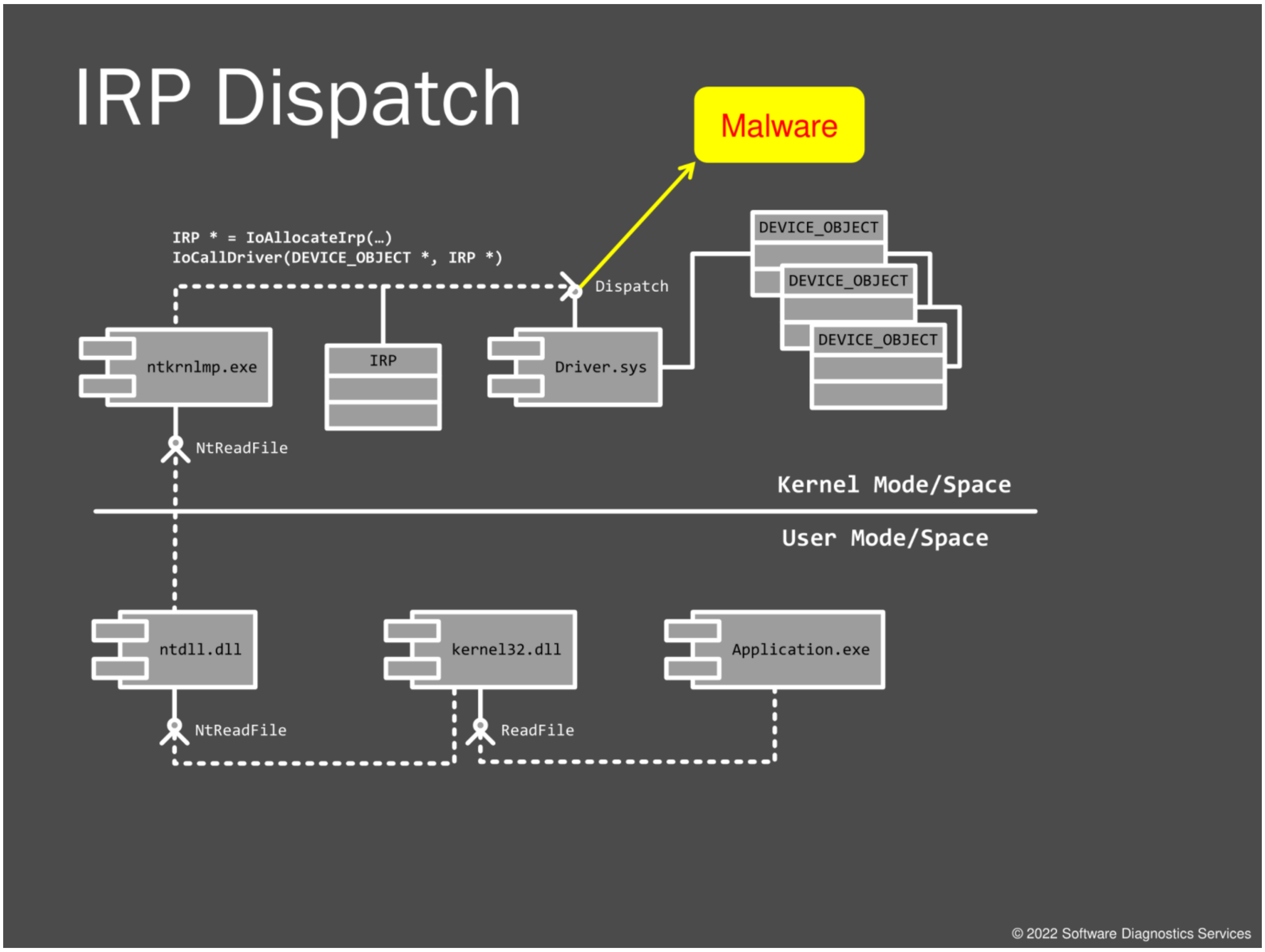
SSDT

System Service Dispatch Table

```
1: kd> uf ntdll!NtReadFIle
ntdll!NtReadFile:
00007ffe`5b023800 4c8bd1           mov     r10,rcx
00007ffe`5b023803 b806000000       mov     eax,6
00007ffe`5b023808 f604250803fe7f01 test    byte ptr [SharedUserData+0x308 (00000000`7ffe0308)],1
00007ffe`5b023810 7503             jne     ntdll!NtReadFile+0x15 (00007ffe`5b023815)  Branch

ntdll!NtReadFile+0x12:
00007ffe`5b023812 0f05             syscall
00007ffe`5b023814 c3               ret

ntdll!NtReadFile+0x15:
00007ffe`5b023815 cd2e             int     2Eh
00007ffe`5b023817 c3               ret
```

```
1: kd> u nt!KiServiceTable + (dwo(nt!KiServiceTable+4*6) >>> 4) L1
nt!NtReadFile:
fffff807`62780750 4c8bdc           mov     r11,rsp
```

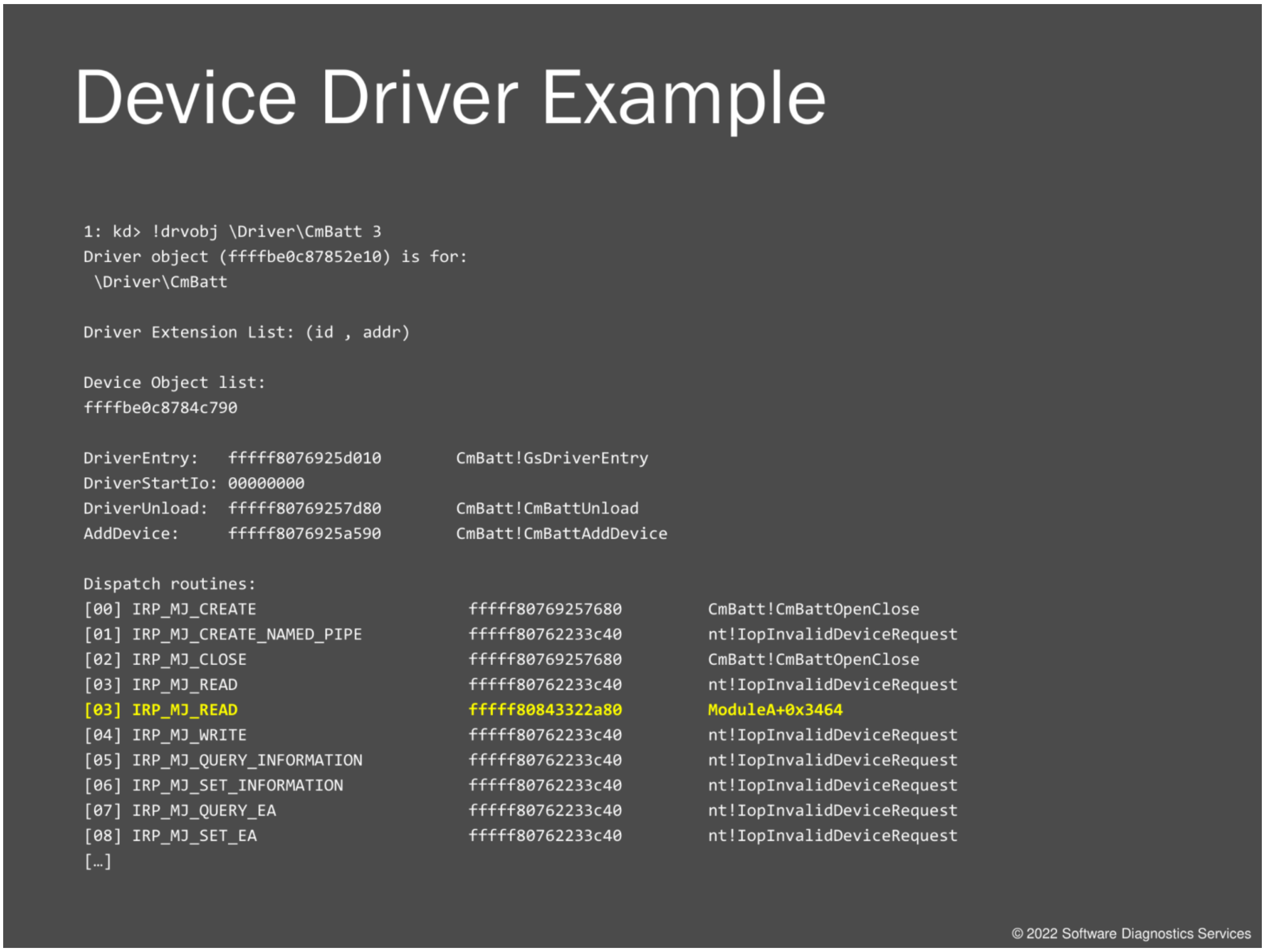
User space calls from DLLs such as *user32*, *gdi32*, and *kernel32* are forwarded to the *ntdll* module from which they transition to kernel space. The kernel maintains a special table containing pointers to corresponding kernel functions. In this slide, for example, we see the *ReadFile* API call is mapped to the 6th entry in the service table. This table can be hooked too, and the presence of any raw pointers or pointers to code outside the nt module range should trigger suspicion. The example here is from the 64-bit Windows SSDT. On the 32-bit Windows system, SSDT is simpler, and I show you that too.



IRP Dispatch

```
IRP * = IoAllocateIrp(…)
IoCallDriver(DEVICE_OBJECT *, IRP *)
```

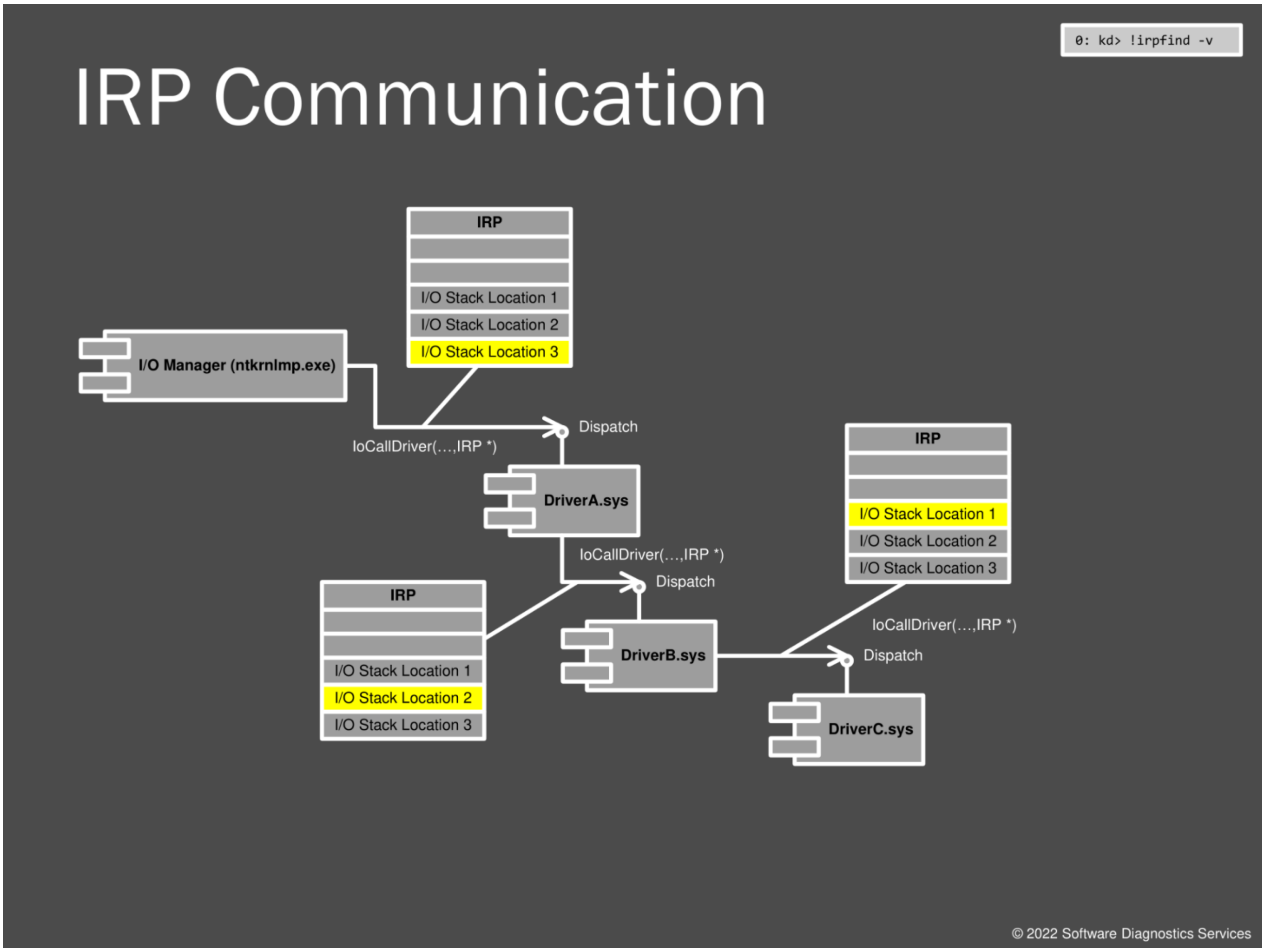
This slide shows a big picture of I/O. Requests such as reading and writing to a device are implemented by a packet-driven architecture. Upon such a request, I/O Manager (a loosely defined component in kernel space) allocates a structure to describe a request, including pointers to buffers for device data, and then passes it through the device driver stack (for example, file system -> volume -> disk array -> disk). Notice that an IRP is created and passed to Driver.sys code. There, according to an IRP dispatch table, an appropriate function is called. This table can be hooked by malware.



Device Driver Example

```
1: kd> !drvobj \Driver\CmBatt 3
Driver object (ffffbe0c87852e10) is for:
 \Driver\CmBatt

Driver Extension List: (id , addr)

Device Object list:
ffffbe0c8784c790

DriverEntry:   fffff8076925d010       CmBatt!GsDriverEntry
DriverStartIo: 00000000
DriverUnload:  fffff80769257d80       CmBatt!CmBattUnload
AddDevice:     fffff8076925a590       CmBatt!CmBattAddDevice

Dispatch routines:
[00] IRP_MJ_CREATE                      fffff80769257680        CmBatt!CmBattOpenClose
[01] IRP_MJ_CREATE_NAMED_PIPE           fffff80762233c40        nt!IopInvalidDeviceRequest
[02] IRP_MJ_CLOSE                       fffff80769257680        CmBatt!CmBattOpenClose
[03] IRP_MJ_READ                        fffff80762233c40        nt!IopInvalidDeviceRequest
[03] IRP_MJ_READ                        fffff80843322a80        ModuleA+0x3464
[04] IRP_MJ_WRITE                       fffff80762233c40        nt!IopInvalidDeviceRequest
[05] IRP_MJ_QUERY_INFORMATION           fffff80762233c40        nt!IopInvalidDeviceRequest
[06] IRP_MJ_SET_INFORMATION             fffff80762233c40        nt!IopInvalidDeviceRequest
[07] IRP_MJ_QUERY_EA                    fffff80762233c40        nt!IopInvalidDeviceRequest
[08] IRP_MJ_SET_EA                      fffff80762233c40        nt!IopInvalidDeviceRequest
[…]
```

Here’s a typical device driver example with an IRP dispatch table. Notice a hooked entry there.



```
0: kd> !irpfind -v
```

IRP Communication

I/O Manager (ntkrnlmp.exe)

IRP
I/O Stack Location 1
I/O Stack Location 2
I/O Stack Location 3

IoCallDriver(…,IRP *)

Dispatch

DriverA.sys

IoCallDriver(…,IRP *)

Dispatch

IRP
I/O Stack Location 1
I/O Stack Location 2
I/O Stack Location 3

DriverB.sys

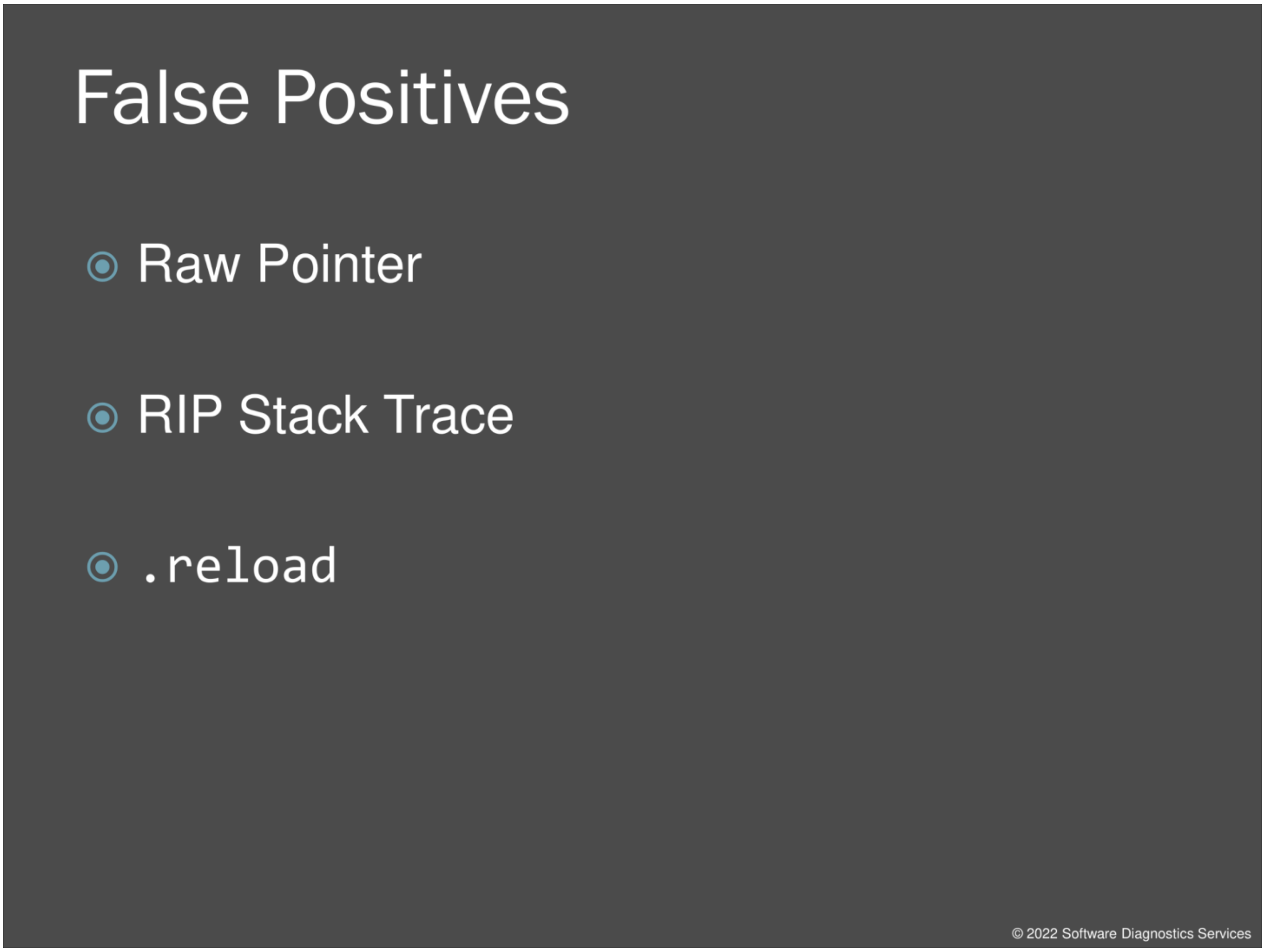
IRP
I/O Stack Location 1
I/O Stack Location 2
I/O Stack Location 3

IoCallDriver(…,IRP *)

Dispatch

DriverC.sys

To keep track of the current device driver in the device driver stack, each I/O Request Packet (IRP) contains a stack at the end of its structure. It is implemented similarly to a thread stack: its pointer (slot index) is decremented from bottom to top. We can dump all such I/O stacks and look for any anomalies.

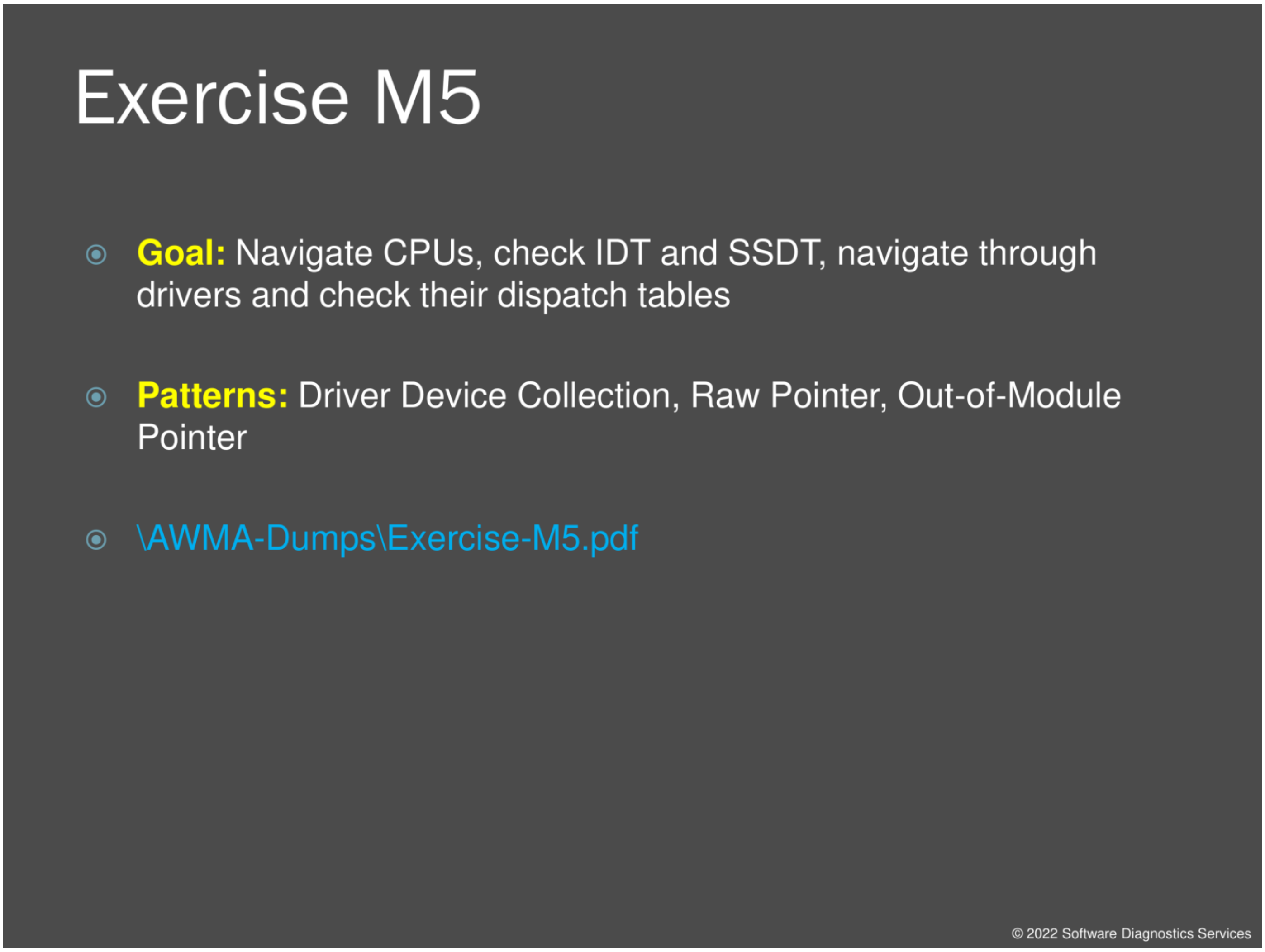


False Positives

- Raw Pointer
- RIP Stack Trace
- `.reload`

© 2022 Software Diagnostics Services

Just before we continue with our next exercise, I would like to mention the possible occurrence of raw pointers or strange references outside the expected range. These might be false positives due to the recent change of context, and we should first try to resolve symbols by the **.reload** command.



Exercise M5

- **Goal:** Navigate CPUs, check IDT and SSDT, navigate through drivers and check their dispatch tables
- **Patterns:** Driver Device Collection, Raw Pointer, Out-of-Module Pointer
- \AWMA-Dumps\Exercise-M5.pdf

Now we analyze a 32-bit complete memory dump.

Exercise M5

Goal: Navigate CPUs, check IDT and SSDT, navigate through drivers and check their dispatch tables.

Patterns: Driver Device Collection, Raw Pointer, Out-of-Module Pointer.

1. Launch WinDbg Preview.

2. Open \AWMA-Dumps\Complete\MEMORY2.DMP.

3. You get the dump file loaded:

```
Microsoft (R) Windows Debugger Version 10.0.25136.1001 AMD64
Copyright (c) Microsoft Corporation. All rights reserved.


Loading Dump File [C:\AWMA-Dumps\Complete\MEMORY2.DMP]
Kernel Complete Dump File: Full address space is available


************* Path validation summary **************
Response                         Time (ms)     Location
Deferred                                       srv*
Symbol search path is: srv*
Executable search path is:
VirtualToOffset: 8b500000 not properly sign extended
Windows Vista Kernel Version 6000 MP (2 procs) Free x86 compatible
Product: WinNt, suite: TerminalServer SingleUserTS Personal
Edition build lab: 6000.16386.x86fre.vista_rtm.061101-2205
Machine Name:
Kernel base = 0x81800000 PsLoadedModuleList = 0x81911db0
Debug session time: Wed Jul 20 22:26:14.859 2011 (UTC + 1:00)
System Uptime: 0 days 0:15:30.657
VirtualToOffset: 90800000 not properly sign extended
Loading Kernel Symbols
...............................................................
................................................................
...........
Loading User Symbols
...................................
Loading unloaded module list
.........VirtualToOffset: bce00000 not properly sign extended
Unable to enumerate user-mode unloaded modules, NTSTATUS 0xC0000147
For analysis of this file, run !analyze -v
eax=818f483c ebx=876a72a0 ecx=000007c8 edx=819293dc esi=818f4820 edi=876a72a0
eip=818d85c9 esp=9377fcb0 ebp=9377fccc iopl=0         nv up ei ng nz na pe nc
cs=0008  ss=0010  ds=0023  es=0023  fs=0030  gs=0000             efl=00000286
nt!KeBugCheckEx+0x1e:
818d85c9 8be5            mov     esp,ebp
```

4. Open a log file:

```
0: kd> .logopen C:\AWMA-Dumps\M5.log
Opened log file 'C:\AWMA-Dumps\M5.log'
```

5. We switch to the second CPU using the **~<*n*>s** command and check its IDT:

```
0: kd> ~1s

1: kd> k
 # ChildEBP RetAddr
WARNING: Frame IP not in any known module. Following frames may be wrong.
00 0018fd8c 7787027f     0xab76be
01 0018fd90 00ab7690     ntdll!NtSecureConnectPort+0xb
02 0018fda8 00ab13fc     0xab7690
VirtualToOffset: bd840000 not properly sign extended
03 0018fdf0 76113833     0xab13fc
04 0018fdfc 7784a9bd     kernel32!BaseThreadInitThunk+0xe
05 0018fe3c 00000000     ntdll!_RtlUserThreadStart+0x23
```

Note: Messages VirtualToOffset: bd840000 not properly sign extended may disappear if you repeat the same command.

It looks like we have a false positive instance of the **RIP Stack Trace** pattern because it disappears as soon as we reload symbols:

```
1: kd> .reload
VirtualToOffset: c0800000 not properly sign extended
Loading Kernel Symbols
...............................................................
................................................................
...........
Loading User Symbols
...
Loading unloaded module list
.........Unable to enumerate user-mode unloaded modules, NTSTATUS 0xC0000147
VirtualToOffset: bced0000 not properly sign extended
Unable to load image C:\Examples\ApplicationE\Release\ApplicationE.exe, Win32 error 0n2

************* Symbol Loading Error Summary **************
Module name            Error
ApplicationE           The system cannot find the file specified

You can troubleshoot most symbol related issues by turning on symbol loading diagnostics (!sym
noisy) and repeating the command that caused symbols to be loaded.
You should also verify that your symbol search path (.sympath) is correct.

1: kd> k
# ChildEBP RetAddr
WARNING: Stack unwind information not available. Following frames may be wrong.
00 0018fda8 00ab13fc     ApplicationE+0x76be
VirtualToOffset: bd820000 not properly sign extended
01 0018fdf0 76113833     ApplicationE+0x13fc
02 0018fdfc 7784a9bd     kernel32!BaseThreadInitThunk+0xe
03 0018fe3c 00000000     ntdll!_RtlUserThreadStart+0x23

1: kd> k
# ChildEBP RetAddr
WARNING: Stack unwind information not available. Following frames may be wrong.
00 0018fda8 00ab13fc     ApplicationE+0x76be
01 0018fdf0 76113833     ApplicationE+0x13fc
02 0018fdfc 7784a9bd     kernel32!BaseThreadInitThunk+0xe
03 0018fe3c 00000000     ntdll!_RtlUserThreadStart+0x23
```

6. Let’s check CPU 1 IDT (we repeat twice to remove VirtualToOffset messages for clarity):

```
1: kd> !idt
[...]

1: kd> !idt

Dumping IDT: 857ee960

37:    81bb50e8 hal!PicSpuriousService37
50:    8393aa50 pci!ExpressRootPortMessageRoutine (KINTERRUPT 8393aa00)

51:    848e37d0 serial!SerialCIsrSw (KINTERRUPT 848e3780)

52:    83951cd0 pci!ExpressRootPortMessageRoutine (KINTERRUPT 83951c80)

53:    8395ca50 pci!ExpressRootPortMessageRoutine (KINTERRUPT 8395ca00)

54:    8399d7d0 pci!ExpressRootPortMessageRoutine (KINTERRUPT 8399d780)

55:    839ac550 ataport!IdePortInterrupt (KINTERRUPT 839ac500)

60:    8393acd0 pci!ExpressRootPortMessageRoutine (KINTERRUPT 8393ac80)

62:    8393a050 pci!ExpressRootPortMessageRoutine (KINTERRUPT 8393a000)

63:    8395ccd0 pci!ExpressRootPortMessageRoutine (KINTERRUPT 8395cc80)

64:    8399da50 pci!ExpressRootPortMessageRoutine (KINTERRUPT 8399da00)

65:    839ac7d0 pci!ExpressRootPortMessageRoutine (KINTERRUPT 839ac780)

70:    83911050 pci!ExpressRootPortMessageRoutine (KINTERRUPT 83911000)

71:    848e3a50 i8042prt!I8042MouseInterruptService (KINTERRUPT 848e3a00)

72:    8393a2d0 pci!ExpressRootPortMessageRoutine (KINTERRUPT 8393a280)

73:    83951050 pci!ExpressRootPortMessageRoutine (KINTERRUPT 83951000)

74:    8399dcd0 pci!ExpressRootPortMessageRoutine (KINTERRUPT 8399dc80)

75:    839aca50 pci!ExpressRootPortMessageRoutine (KINTERRUPT 839aca00)

76:    8764dcd0 ndis!ndisMiniportIsr (KINTERRUPT 8764dc80)

80:    839112d0 pci!ExpressRootPortMessageRoutine (KINTERRUPT 83911280)

81:    848e3cd0 i8042prt!I8042KeyboardInterruptService (KINTERRUPT 848e3c80)

82:    8393a550 pci!ExpressRootPortMessageRoutine (KINTERRUPT 8393a500)

83:    839512d0 pci!ExpressRootPortMessageRoutine (KINTERRUPT 83951280)

84:    8395c050 pci!ExpressRootPortMessageRoutine (KINTERRUPT 8395c000)

85:    839accd0 pci!ExpressRootPortMessageRoutine (KINTERRUPT 839acc80)

86:    848e3050 USBPORT!USBPORT_InterruptService (KINTERRUPT 848e3000)
```

```
90:    83911550 pci!ExpressRootPortMessageRoutine (KINTERRUPT 83911500)

92:    8393a7d0 pci!ExpressRootPortMessageRoutine (KINTERRUPT 8393a780)

93:    83951550 pci!ExpressRootPortMessageRoutine (KINTERRUPT 83951500)

94:    8395c2d0 pci!ExpressRootPortMessageRoutine (KINTERRUPT 8395c280)

95:    8399d050 pci!ExpressRootPortMessageRoutine (KINTERRUPT 8399d000)

96:    848e32d0 vmci!DllUnload+0x552 (KINTERRUPT 848e3280)

                portcls!KspShellTransferKsIrp+0x2a (KINTERRUPT 8764da00)

                dxgkrnl!DpiFdoLineInterruptRoutine (KINTERRUPT 8764d500)

a0:    839117d0 pci!ExpressRootPortMessageRoutine (KINTERRUPT 83911780)

a3:    839517d0 pci!ExpressRootPortMessageRoutine (KINTERRUPT 83951780)

a4:    8395c550 pci!ExpressRootPortMessageRoutine (KINTERRUPT 8395c500)

a5:    8399d2d0 pci!ExpressRootPortMessageRoutine (KINTERRUPT 8399d280)

a6:    839ac050 storport!RaidpAdapterInterruptRoutine (KINTERRUPT 839ac000)

                USBPORT!USBPORT_InterruptService (KINTERRUPT 8764d780)

b0:    83911a50 pci!ExpressRootPortMessageRoutine (KINTERRUPT 83911a00)

b1:    83911cd0 acpi!ACPIInterruptServiceRoutine (KINTERRUPT 83911c80)

b2:    848e3550 serial!SerialCIsrSw (KINTERRUPT 848e3500)

b3:    83951a50 pci!ExpressRootPortMessageRoutine (KINTERRUPT 83951a00)

b4:    8395c7d0 pci!ExpressRootPortMessageRoutine (KINTERRUPT 8395c780)

b5:    8399d550 pci!ExpressRootPortMessageRoutine (KINTERRUPT 8399d500)

b6:    839ac2d0 ataport!IdePortInterrupt (KINTERRUPT 839ac280)

c1:    81bb53d8 hal!HalpBroadcastCallService
d1:    81ba497c hal!HalpClockInterruptPn
df:    81bb51c0 hal!HalpApicRebootService
e1:    81bb5934 hal!HalpIpiHandler
e3:    81bb56d4 hal!HalpLocalApicErrorService
fd:    81bb5edc hal!HalpProfileInterrupt
fe:    81bb6148 hal!HalpPerfInterrupt
ff:    87fe9724 E1G60I32!⍰ntoskrnl_NULL_THUNK_DATA
```

Note that the last entry **ff** differs from expected *hal* and other hardware modules. We check the address of the interrupt function:

```
1: kd> u 87fe9724
VirtualToOffset: 87fe9724 not properly sign extended
87fe9724 0000            add     byte ptr [eax],al
VirtualToOffset: 87fe9726 not properly sign extended
87fe9726 0000            add     byte ptr [eax],al
```

```
VirtualToOffset: 87fe9728 not properly sign extended
87fe9728 0000            add     byte ptr [eax],al
VirtualToOffset: 87fe972a not properly sign extended
87fe972a 0000            add     byte ptr [eax],al
VirtualToOffset: 87fe972c not properly sign extended
87fe972c 0000            add     byte ptr [eax],al
VirtualToOffset: 87fe972e not properly sign extended
87fe972e 0000            add     byte ptr [eax],al
VirtualToOffset: 87fe9730 not properly sign extended
87fe9730 0000            add     byte ptr [eax],al
VirtualToOffset: 87fe9732 not properly sign extended
87fe9732 0000            add     byte ptr [eax],al

1: kd> u
VirtualToOffset: 87fe9734 not properly sign extended
87fe9734 db6ad2          fld     tbyte ptr [edx-2Eh]
VirtualToOffset: 87fe9737 not properly sign extended
87fe9737 44              inc     esp
VirtualToOffset: 87fe9738 not properly sign extended
87fe9738 0000            add     byte ptr [eax],al
VirtualToOffset: 87fe973a not properly sign extended
87fe973a 0000            add     byte ptr [eax],al
VirtualToOffset: 87fe973c not properly sign extended
87fe973c 0200            add     al,byte ptr [eax]
VirtualToOffset: 87fe973e not properly sign extended
87fe973e 0000            add     byte ptr [eax],al
VirtualToOffset: 87fe9740 not properly sign extended
87fe9740 25000000c0      and     eax,0C0000000h
VirtualToOffset: 87fe9745 not properly sign extended
87fe9745 58              pop     eax
```

The code seems wild, and most likely, if some code uses this interrupt for communication, it definitely crashes the system. On the other hand, the module itself seems normal as it has symbol files, and we hypothesize it was modified by malware to hide malicious activities under its name, but something went wrong with hooking IDT.

7. We now check SSDT. To dump it, we need to know its size:

```
1: kd> dps nt!KeServiceDescriptorTable
[...]

1: kd> dps nt!KeServiceDescriptorTable
81931b00  81880624 nt!KiServiceTable
81931b04  00000000
81931b08  0000018e
81931b0c  81880c60 nt!KiArgumentTable
81931b10  00000000
81931b14  00000000
81931b18  00000000
81931b1c  00000000
81931b20  00000021
81931b24  82b85ad0
81931b28  e57a42bd
81931b2c  d6bf94d5
81931b30  00000200
81931b34  82b81910
81931b38  00000000
81931b3c  00000000
81931b40  81880624 nt!KiServiceTable
81931b44  00000000
```

```
81931b48  0000018e
81931b4c  81880c60 nt!KiArgumentTable
81931b50  8a9ca000 win32k!W32pServiceTable
81931b54  00000000
81931b58  00000304
81931b5c  8a9caf20 win32k!W32pArgumentTable
81931b60  82b817a0
81931b64  82b81350
81931b68  82b81630
81931b6c  82b814c0
81931b70  00000000
81931b74  82b811e0
81931b78  00000000
81931b7c  00000000

1: kd> dps nt!KiServiceTable L18e
[...]

1: kd> dps nt!KiServiceTable L18e
81880624  819be057 nt!NtAcceptConnectPort
81880628  818657ce nt!NtAccessCheck
8188062c  81a4a707 nt!NtAccessCheckAndAuditAlarm
81880630  81865805 nt!NtAccessCheckByType
81880634  81a4a746 nt!NtAccessCheckByTypeAndAuditAlarm
81880638  81865840 nt!NtAccessCheckByTypeResultList
8188063c  81a4a78f nt!NtAccessCheckByTypeResultListAndAuditAlarm
81880640  81a4a7d8 nt!NtAccessCheckByTypeResultListAndAuditAlarmByHandle
81880644  81a88f47 nt!NtAddAtom
81880648  81a8aff4 nt!NtAddBootEntry
8188064c  81a8c282 nt!NtAddDriverEntry
81880650  81a3eee5 nt!NtAdjustGroupsToken
81880654  81a3eacd nt!NtAdjustPrivilegesToken
81880658  81a1d327 nt!NtAlertResumeThread
8188065c  81a1d2cf nt!NtAlertThread
81880660  81a89390 nt!NtAllocateLocallyUniqueId
81880664  819e743f nt!NtAllocateUserPhysicalPages
81880668  81a88a70 nt!NtAllocateUuids
8188066c  819d531f nt!NtAllocateVirtualMemory
81880670  819c0b37 nt!NtAlpcAcceptConnectPort
81880674  819c62c7 nt!NtAlpcCancelMessage
81880678  819bfe3b nt!NtAlpcConnectPort
8188067c  819bf54b nt!NtAlpcCreatePort
81880680  819c839b nt!NtAlpcCreatePortSection
81880684  819c9cc3 nt!NtAlpcCreateResourceReserve
81880688  819c8637 nt!NtAlpcCreateSectionView
8188068c  819ca27f nt!NtAlpcCreateSecurityContext
81880690  819c853a nt!NtAlpcDeletePortSection
81880694  819c9dfa nt!NtAlpcDeleteResourceReserve
81880698  819c886d nt!NtAlpcDeleteSectionView
8188069c  819ca577 nt!NtAlpcDeleteSecurityContext
818806a0  819cc39b nt!NtAlpcDisconnectPort
818806a4  819ca803 nt!NtAlpcImpersonateClientOfPort
818806a8  819ce107 nt!NtAlpcOpenSenderProcess
818806ac  819ce6b7 nt!NtAlpcOpenSenderThread
818806b0  819cd953 nt!NtAlpcQueryInformation
818806b4  819c70d5 nt!NtAlpcQueryInformationMessage
818806b8  819ca430 nt!NtAlpcRevokeSecurityContext
818806bc  819c615b nt!NtAlpcSendWaitReceivePort
818806c0  819cd48b nt!NtAlpcSetInformation
818806c4  81a9f2f9 nt!NtApphelpCacheControl
```

```
818806c8  819d21cb nt!NtAreMappedFilesTheSame
818806cc  81a1f5bb nt!NtAssignProcessToJobObject
818806d0  8188037c nt!NtCallbackReturn
818806d4  8198046c nt!NtRequestDeviceWakeup
818806d8  8198bd6c nt!NtCancelIoFile
818806dc  81879318 nt!NtCancelTimer
818806e0  81a87095 nt!NtClearEvent
818806e4  819f189c nt!NtClose
818806e8  81a4acc9 nt!NtCloseObjectAuditAlarm
818806ec  8193cd2b nt!NtCompactKeys
818806f0  81a4e0c9 nt!NtCompareTokens
818806f4  819be0db nt!NtCompleteConnectPort
818806f8  8193cfb7 nt!NtCompressKey
818806fc  819be023 nt!NtConnectPort
81880700  818903b8 nt!NtContinue
81880704  819752d2 nt!NtCreateDebugObject
81880708  819ed9df nt!NtCreateDirectoryObject
8188070c  81a870e8 nt!NtCreateEvent
81880710  81a8fa91 nt!NtCreateEventPair
81880714  8198ec5e nt!NtCreateFile
81880718  8198b298 nt!NtCreateIoCompletion
8188071c  81a1f339 nt!NtCreateJobObject
81880720  81a2210f nt!NtCreateJobSet
81880724  81937576 nt!NtCreateKey
81880728  819375d9 nt!NtCreateKeyTransacted
8188072c  8198ed8f nt!NtCreateMailslotFile
81880730  81a8ff0a nt!NtCreateMutant
81880734  8198eca1 nt!NtCreateNamedPipeFile
81880738  819fa0b6 nt!NtCreatePrivateNamespace
8188073c  819e37ec nt!NtCreatePagingFile
81880740  819bdb25 nt!NtCreatePort
81880744  81a123b2 nt!NtCreateProcess
81880748  81a123fd nt!NtCreateProcessEx
8188074c  81a90403 nt!NtCreateProfile
81880750  819d7703 nt!NtCreateSection
81880754  81a880ff nt!NtCreateSemaphore
81880758  819efc6b nt!NtCreateSymbolicLinkObject
8188075c  81a11f31 nt!NtCreateThread
81880760  81a8f6f1 nt!NtCreateTimer
81880764  81a4cced nt!NtCreateToken
81880768  81a53ac4 nt!NtCreateTransaction
8188076c  81a53dd7 nt!NtOpenTransaction
81880770  81a53fcf nt!NtQueryInformationTransaction
81880774  81a56472 nt!NtQueryInformationTransactionManager
81880778  81a54e64 nt!NtPrePrepareEnlistment
8188077c  81a54da3 nt!NtPrepareEnlistment
81880780  81a54f25 nt!NtCommitEnlistment
81880784  81a553a9 nt!NtReadOnlyEnlistment
81880788  81a55468 nt!NtRollbackComplete
8188078c  81a54fe6 nt!NtRollbackEnlistment
81880790  81a544cf nt!NtCommitTransaction
81880794  81a54538 nt!NtRollbackTransaction
81880798  81a55168 nt!NtPrePrepareComplete
8188079c  81a550a7 nt!NtPrepareComplete
818807a0  81a55229 nt!NtCommitComplete
818807a4  81a552ea nt!NtSinglePhaseReject
818807a8  81a545b5 nt!NtSetInformationTransaction
818807ac  81a56879 nt!NtSetInformationTransactionManager
818807b0  81a55d36 nt!NtSetInformationResourceManager
818807b4  81a55ed0 nt!NtCreateTransactionManager
```

```
818807b8  81a560e7 nt!NtOpenTransactionManager
818807bc  81a56356 nt!NtRollforwardTransactionManager
818807c0  81a549c3 nt!NtRecoverEnlistment
818807c4  81a55999 nt!NtRecoverResourceManager
818807c8  81a56417 nt!NtRecoverTransactionManager
818807cc  81a55527 nt!NtCreateResourceManager
818807d0  81a557ed nt!NtOpenResourceManager
818807d4  81a559f2 nt!NtGetNotificationResourceManager
818807d8  81a55b07 nt!NtQueryInformationResourceManager
818807dc  81a5470d nt!NtCreateEnlistment
818807e0  81a547fa nt!NtOpenEnlistment
818807e4  81a54c06 nt!NtSetInformationEnlistment
818807e8  81a54a1f nt!NtQueryInformationEnlistment
818807ec  81a89383 nt!NtStartTm
818807f0  819bdb8f nt!NtCreateWaitablePort
818807f4  81976096 nt!NtDebugActiveProcess
818807f8  819766ec nt!NtDebugContinue
818807fc  81a90aa5 nt!NtDelayExecution
81880800  81a891fb nt!NtDeleteAtom
81880804  81a8b027 nt!NtDeleteBootEntry
81880808  81a8c2b3 nt!NtDeleteDriverEntry
8188080c  8198c187 nt!NtDeleteFile
81880810  819379a7 nt!NtDeleteKey
81880814  819fa6aa nt!NtDeletePrivateNamespace
81880818  81a4adab nt!NtDeleteObjectAuditAlarm
8188081c  81937c3a nt!NtDeleteValueKey
81880820  8198ee63 nt!NtDeviceIoControlFile
81880824  81a7a099 nt!NtDisplayString
81880828  819f1fb3 nt!NtDuplicateObject
8188082c  81a3f88b nt!NtDuplicateToken
81880830  81a8b228 nt!NtEnumerateBootEntries
81880834  81a8c4b2 nt!NtEnumerateDriverEntries
81880838  81937f12 nt!NtEnumerateKey
8188083c  81a8adfb nt!NtEnumerateSystemEnvironmentValuesEx
81880840  81868f61 nt!NtEnumerateTransactionObject
81880844  81938171 nt!NtEnumerateValueKey
81880848  819e1387 nt!NtExtendSection
8188084c  81a40316 nt!NtFilterToken
81880850  81a890a1 nt!NtFindAtom
81880854  8198c299 nt!NtFlushBuffersFile
81880858  819e84b3 nt!NtFlushInstructionCache
8188085c  819383f0 nt!NtFlushKey
81880860  818cdfab nt!NtFlushProcessWriteBuffers
81880864  819da8e1 nt!NtFlushVirtualMemory
81880868  819e84a0 nt!NtFlushWriteBuffer
8188086c  819e7b6e nt!NtFreeUserPhysicalPages
81880870  818beb63 nt!NtFreeVirtualMemory
81880874  818d0683 nt!NtFreezeRegistry
81880878  81869169 nt!NtFreezeTransactions
8188087c  8198ee9f nt!NtFsControlFile
81880880  81a1a9bf nt!NtGetContextThread
81880884  81a0dbc7 nt!NtGetDevicePowerState
81880888  81a8610b nt!NtGetNlsSectionPtr
8188088c  819b9d7a nt!NtGetPlugPlayEvent
81880890  818e4864 nt!NtGetWriteWatch
81880894  81a4decf nt!NtImpersonateAnonymousToken
81880898  819be383 nt!NtImpersonateClientOfPort
8188089c  81a22455 nt!NtImpersonateThread
818808a0  81a84da7 nt!NtInitializeNlsFiles
818808a4  8193860d nt!NtInitializeRegistry
```

```
818808a8  81a0d9b8 nt!NtInitiatePowerAction
818808ac  81a21f63 nt!NtIsProcessInJob
818808b0  81a0dbad nt!NtIsSystemResumeAutomatic
818808b4  819be3b1 nt!NtListenPort
818808b8  81998384 nt!NtLoadDriver
818808bc  8193a414 nt!NtLoadKey
818808c0  8193a43b nt!NtLoadKey2
818808c4  8193a467 nt!NtLoadKeyEx
818808c8  8198eedb nt!NtLockFile
818808cc  81a7a35c nt!NtLockProductActivationKeys
818808d0  8193d08e nt!NtLockRegistryKey
818808d4  8181ad7f nt!NtLockVirtualMemory
818808d8  819ef3b9 nt!NtMakePermanentObject
818808dc  819f18cb nt!NtMakeTemporaryObject
818808e0  819e67e2 nt!NtMapUserPhysicalPages
818808e4  819e6d4b nt!NtMapUserPhysicalPagesScatter
818808e8  819d0206 nt!NtMapViewOfSection
818808ec  81a8b1f7 nt!NtModifyBootEntry
818808f0  81a8c483 nt!NtModifyDriverEntry
818808f4  8198fd76 nt!NtNotifyChangeDirectoryFile
818808f8  81938716 nt!NtNotifyChangeKey
818808fc  81938753 nt!NtNotifyChangeMultipleKeys
81880900  819edae3 nt!NtOpenDirectoryObject
81880904  81a87211 nt!NtOpenEvent
81880908  81a8fbc7 nt!NtOpenEventPair
8188090c  819900cb nt!NtOpenFile
81880910  8198b3a5 nt!NtOpenIoCompletion
81880914  81a1f4f7 nt!NtOpenJobObject
81880918  8193922f nt!NtOpenKey
8188091c  8193928b nt!NtOpenKeyTransacted
81880920  81a9000f nt!NtOpenMutant
81880924  819fa335 nt!NtOpenPrivateNamespace
81880928  81a4a823 nt!NtOpenObjectAuditAlarm
8188092c  81a1385d nt!NtOpenProcess
81880930  81a40d3c nt!NtOpenProcessToken
81880934  81a40d61 nt!NtOpenProcessTokenEx
81880938  819da58b nt!NtOpenSection
8188093c  81a8822b nt!NtOpenSemaphore
81880940  819e46cf nt!NtOpenSession
81880944  819efe95 nt!NtOpenSymbolicLinkObject
81880948  81a13bbf nt!NtOpenThread
8188094c  81a40f2b nt!NtOpenThreadToken
81880950  81a40f53 nt!NtOpenThreadTokenEx
81880954  81a8f840 nt!NtOpenTimer
81880958  819b9eff nt!NtPlugPlayControl
8188095c  81a079bc nt!NtPowerInformation
81880960  81a4fd36 nt!NtPrivilegeCheck
81880964  81a49869 nt!NtPrivilegeObjectAuditAlarm
81880968  81a49aca nt!NtPrivilegedServiceAuditAlarm
8188096c  819e8767 nt!NtProtectVirtualMemory
81880970  81a872e4 nt!NtPulseEvent
81880974  8198c4b5 nt!NtQueryAttributesFile
81880978  81a8b6d3 nt!NtQueryBootEntryOrder
8188097c  81a8bb27 nt!NtQueryBootOptions
81880980  8187c403 nt!NtQueryDebugFilterState
81880984  81a7ec28 nt!NtQueryDefaultLocale
81880988  81a7efaf nt!NtQueryDefaultUILanguage
8188098c  8198fd0d nt!NtQueryDirectoryFile
81880990  819edba2 nt!NtQueryDirectoryObject
81880994  81a8c03b nt!NtQueryDriverEntryOrder
```

```
81880998  81990107 nt!NtQueryEaFile
8188099c  81a873c7 nt!NtQueryEvent
818809a0  8198c657 nt!NtQueryFullAttributesFile
818809a4  81a89228 nt!NtQueryInformationAtom
818809a8  81990cf6 nt!NtQueryInformationFile
818809ac  81a1ff3f nt!NtQueryInformationJobObject
818809b0  819be429 nt!NtQueryInformationPort
818809b4  81a14191 nt!NtQueryInformationProcess
818809b8  81a1774b nt!NtQueryInformationThread
818809bc  81a41198 nt!NtQueryInformationToken
818809c0  81a7ef2b nt!NtQueryInstallUILanguage
818809c4  81a908f7 nt!NtQueryIntervalProfile
818809c8  8198b47c nt!NtQueryIoCompletion
818809cc  81939557 nt!NtQueryKey
818809d0  8193be73 nt!NtQueryMultipleValueKey
818809d4  81a900e2 nt!NtQueryMutant
818809d8  819f7c1d nt!NtQueryObject
818809dc  8193c4e7 nt!NtQueryOpenSubKeys
818809e0  8193c76b nt!NtQueryOpenSubKeysEx
818809e4  81a909b0 nt!NtQueryPerformanceCounter
818809e8  819920e7 nt!NtQueryQuotaInformationFile
818809ec  819e34f2 nt!NtQuerySection
818809f0  819f470b nt!NtQuerySecurityObject
818809f4  81a882fe nt!NtQuerySemaphore
818809f8  819eff54 nt!NtQuerySymbolicLinkObject
818809fc  81a8a223 nt!NtQuerySystemEnvironmentValue
81880a00  81a8a831 nt!NtQuerySystemEnvironmentValueEx
81880a04  889aa114 crashdmp!⍰ntoskrnl_NULL_THUNK_DATA
81880a08  81a7ac06 nt!NtQuerySystemTime
81880a0c  81a8f913 nt!NtQueryTimer
81880a10  81a7aeeb nt!NtQueryTimerResolution
81880a14  8193985a nt!NtQueryValueKey
81880a18  819e9273 nt!NtQueryVirtualMemory
81880a1c  8199274e nt!NtQueryVolumeInformationFile
81880a20  81a1a655 nt!NtQueueApcThread
81880a24  81890400 nt!NtRaiseException
81880a28  81a87cb7 nt!NtRaiseHardError
81880a2c  8199302b nt!NtReadFile
81880a30  819936b7 nt!NtReadFileScatter
81880a34  819be4e9 nt!NtReadRequestData
81880a38  819d6eee nt!NtReadVirtualMemory
81880a3c  81a1c3c5 nt!NtRegisterThreadTerminatePort
81880a40  81a9028f nt!NtReleaseMutant
81880a44  81a88447 nt!NtReleaseSemaphore
81880a48  8198b61b nt!NtRemoveIoCompletion
81880a4c  819761e1 nt!NtRemoveProcessDebug
81880a50  8193caab nt!NtRenameKey
81880a54  8193bd46 nt!NtReplaceKey
81880a58  819be5c3 nt!NtReplyPort
81880a5c  819be6c8 nt!NtReplyWaitReceivePort
81880a60  819be6ef nt!NtReplyWaitReceivePortEx
81880a64  819be92f nt!NtReplyWaitReplyPort
81880a68  8198046c nt!NtRequestDeviceWakeup
81880a6c  819be253 nt!NtRequestPort
81880a70  819be31c nt!NtRequestWaitReplyPort
81880a74  81a0d95b nt!NtRequestWakeupLatency
81880a78  81a874f7 nt!NtResetEvent
81880a7c  818e5127 nt!NtResetWriteWatch
81880a80  81939bb0 nt!NtRestoreKey
81880a84  81a1d271 nt!NtResumeProcess
```

```
81880a88  81a1d130 nt!NtResumeThread
81880a8c  81939ccf nt!NtSaveKey
81880a90  81939dd6 nt!NtSaveKeyEx
81880a94  81939f21 nt!NtSaveMergedKeys
81880a98  81a579bb nt!NtSavepointComplete
81880a9c  8198046c nt!NtRequestDeviceWakeup
81880aa0  81a579bb nt!NtSavepointComplete
81880aa4  81a545a1 nt!TmSavepointTransaction
81880aa8  81a579bb nt!NtSavepointComplete
81880aac  819bdbf9 nt!NtSecureConnectPort
81880ab0  81a8b91a nt!NtSetBootEntryOrder
81880ab4  81a8be1c nt!NtSetBootOptions
81880ab8  81a1ac4b nt!NtSetContextThread
81880abc  81a9a87b nt!NtSetDebugFilterState
81880ac0  81a88043 nt!NtSetDefaultHardErrorPort
81880ac4  81a7ecaf nt!NtSetDefaultLocale
81880ac8  81a7f995 nt!NtSetDefaultUILanguage
81880acc  81a8c8bd nt!NtSetDriverEntryOrder
81880ad0  8199070d nt!NtSetEaFile
81880ad4  81a875d6 nt!NtSetEvent
81880ad8  81a876bb nt!NtSetEventBoostPriority
81880adc  81a8fea7 nt!NtSetHighEventPair
81880ae0  81a8fdd9 nt!NtSetHighWaitLowEventPair
81880ae4  8197684d nt!NtSetInformationDebugObject
81880ae8  81991555 nt!NtSetInformationFile
81880aec  81a20763 nt!NtSetInformationJobObject
81880af0  8193b8e3 nt!NtSetInformationKey
81880af4  819f82e7 nt!NtSetInformationObject
81880af8  81a15c65 nt!NtSetInformationProcess
81880afc  81a183c7 nt!NtSetInformationThread
81880b00  81a5056f nt!NtSetInformationToken
81880b04  81a908d4 nt!NtSetIntervalProfile
81880b08  8198b5b4 nt!NtSetIoCompletion
81880b0c  81a1eff7 nt!NtSetLdtEntries
81880b10  81a8fe44 nt!NtSetLowEventPair
81880b14  81a8fd6e nt!NtSetLowWaitHighEventPair
81880b18  81992739 nt!NtSetQuotaInformationFile
81880b1c  819f44f0 nt!NtSetSecurityObject
81880b20  81a8a52f nt!NtSetSystemEnvironmentValue
81880b24  81a8ab57 nt!NtSetSystemEnvironmentValueEx
81880b28  81a829f3 nt!NtSetSystemInformation
81880b2c  81ac7bb4 nt!NtSetSystemPowerState
81880b30  81a7acaa nt!NtSetSystemTime
81880b34  81a0d82d nt!NtSetThreadExecutionState
81880b38  818794bf nt!NtSetTimer
81880b3c  81a7afca nt!NtSetTimerResolution
81880b40  81a888eb nt!NtSetUuidSeed
81880b44  8193a08b nt!NtSetValueKey
81880b48  81992c2f nt!NtSetVolumeInformationFile
81880b4c  81a7a057 nt!NtShutdownSystem
81880b50  81847951 nt!NtSignalAndWaitForSingleObject
81880b54  81a90642 nt!NtStartProfile
81880b58  81a90813 nt!NtStopProfile
81880b5c  81a1d213 nt!NtSuspendProcess
81880b60  81a1d047 nt!NtSuspendThread
81880b64  81a90b4f nt!NtSystemDebugControl
81880b68  81a21670 nt!NtTerminateJobObject
81880b6c  81a1b043 nt!NtTerminateProcess
81880b70  81a1b497 nt!NtTerminateThread
81880b74  81a1d42e nt!NtTestAlert
```

```
81880b78  818d06e7 nt!NtThawRegistry
81880b7c  81869250 nt!NtThawTransactions
81880b80  8186e91b nt!NtTraceEvent
81880b84  81a6db67 nt!NtTraceControl
81880b88  81a8cacb nt!NtTranslateFilePath
81880b8c  81998552 nt!NtUnloadDriver
81880b90  8193abd4 nt!NtUnloadKey
81880b94  8193abf3 nt!NtUnloadKey2
81880b98  8193b219 nt!NtUnloadKeyEx
81880b9c  8198f34f nt!NtUnlockFile
81880ba0  81815d20 nt!NtUnlockVirtualMemory
81880ba4  819e0bf0 nt!NtUnmapViewOfSection
81880ba8  81a5c76c nt!NtVdmControl
81880bac  8197642f nt!NtWaitForDebugEvent
81880bb0  819f514c nt!NtWaitForMultipleObjects
81880bb4  819f5027 nt!NtWaitForSingleObject
81880bb8  81a8fd05 nt!NtWaitHighEventPair
81880bbc  81a8fc9c nt!NtWaitLowEventPair
81880bc0  81993c33 nt!NtWriteFile
81880bc4  8199436b nt!NtWriteFileGather
81880bc8  819be556 nt!NtWriteRequestData
81880bcc  819d701b nt!NtWriteVirtualMemory
81880bd0  818b59c6 nt!NtYieldExecution
81880bd4  81a90f41 nt!NtCreateKeyedEvent
81880bd8  81a91073 nt!NtOpenKeyedEvent
81880bdc  81a9114d nt!NtReleaseKeyedEvent
81880be0  81a91434 nt!NtWaitForKeyedEvent
81880be4  81a15902 nt!NtQueryPortInformationProcess
81880be8  81a18eee nt!NtGetCurrentProcessorNumber
81880bec  819f525b nt!NtWaitForMultipleObjects32
81880bf0  81a1d964 nt!NtGetNextProcess
81880bf4  81a1dbd1 nt!NtGetNextThread
81880bf8  8198bf27 nt!NtCancelIoFileEx
81880bfc  8198c064 nt!NtCancelSynchronousIoFile
81880c00  8198b7b4 nt!NtRemoveIoCompletionEx
81880c04  81869663 nt!NtRegisterProtocolAddressInformation
81880c08  81869672 nt!NtPullTransaction
81880c0c  818696af nt!NtMarshallTransaction
81880c10  81869687 nt!NtPropagationComplete
81880c14  8186969b nt!CcTestControl
81880c18  81a9171b nt!NtCreateWorkerFactory
81880c1c  81879c2d nt!NtReleaseWorkerFactoryWorker
81880c20  81879ce4 nt!NtWaitForWorkViaWorkerFactory
81880c24  81879fd7 nt!NtSetInformationWorkerFactory
81880c28  8187a4a7 nt!NtQueryInformationWorkerFactory
81880c2c  8187a72f nt!NtWorkerFactoryWorkerReady
81880c30  81a919be nt!NtShutdownWorkerFactory
81880c34  81a23d84 nt!NtCreateThreadEx
81880c38  81a2256f nt!NtCreateUserProcess
81880c3c  81a7c753 nt!NtQueryLicenseValue
81880c40  81a92b75 nt!NtMapCMFModule
81880c44  81a545a1 nt!TmSavepointTransaction
81880c48  81a9354d nt!NtIsUILanguageComitted
81880c4c  81a9356f nt!NtFlushInstallUILanguage
81880c50  81a9317f nt!NtGetMUIRegistryInfo
81880c54  81a91b88 nt!NtAcquireCMFViewOwnership
81880c58  81a91d4f nt!NtReleaseCMFViewOwnership
```

Note that one of the entries is outside the *nt* module range and points to an address in the *crashdmp* module range.

8. To navigate drivers and their devices which are represented as objects we can use **!object** command:

```
1: kd> !object \Driver
Object: 8585c218  Type: (82b38d60) Directory
    ObjectHeader: 8585c200 (old version)
    HandleCount: 0  PointerCount: 87
    Directory Object: 858074c0  Name: Driver

    Hash Address  Type          Name
    ---- -------  ----          ----
     00  8395e688 Driver        NDIS
         83eaeaf0 Driver        KSecDD
         87746840 Driver        Beep
     01  84beff38 Driver        mouclass
     03  848ea030 Driver        vm3dmp
         848ae9e0 Driver        kbdclass
     04  876a62c8 Driver        monitor
         8392dec0 Driver        msisadrv
         83932688 Driver        Compbatt
         8760a848 Driver        NDProxy
         87768590 Driver        VgaSave
     05  839d6708 Driver        Ecache
         83933688 Driver        MountMgr
     08  87d59128 Driver        PEAUTH
         83993660 Driver        atapi
         848ec2f0 Driver        vmmouse
     09  83937688 Driver        volmgrx
         879e4030 Driver        VMAUDIO
     10  87753590 Driver        RasAcd
         8776c868 Driver        PSched
     11  87738720 Driver        Win32k
         8780b9b0 Driver        usbuhci
         877858c8 Driver        mouhid
     12  877fa410 Driver        usbhub
         84aa5e38 Driver        tunnel
         848e2e08 Driver        swenum
     13  87cd4458 Driver        HTTP
         848c5b30 Driver        RasPppoe
         8774c3e0 Driver        RDPCDD
         877f3910 Driver        usbccgp
     14  848e2c60 Driver        TermDD
     15  848c5030 Driver        fdc
         848ec4e0 Driver        Rasl2tp
     16  87d48268 Driver        Parvdm
     17  879e6f38 Driver        umbus
         848c06b0 Driver        vmci
     18  87d5b560 Driver        secdrv
         82b41190 Driver        ACPI_HAL
         82b37f00 Driver        WMIxWDM
         8395a688 Driver        CLFS
         843271f8 Driver        crcdisk
         84b1ded0 Driver        Serenum
         848e8e30 Driver        PptpMiniport
         8778c630 Driver        Smb
     19  83e4c1c8 Driver        spldr
     21  87d5e368 Driver        tcpipreg
         839d6610 Driver        agp440
         877f3120 Driver        netbt
     22  848bf5a0 Driver        iScsiPrt
         879e6880 Driver        mssmbios
         8780b578 Driver        cdrom
         8760e988 Driver        RDPENCDD
     23  877d7d98 Driver        tdx
         8397fde8 Driver        rspndr
     24  87d2df00 Driver        mpsdrv
         87745608 Driver        Tcpip
     25  83e50f38 Driver        volsnap
         83931688 Driver        volmgr
         877fcf38 Driver        nsiproxy
     26  87668258 Driver        intelppm
     27  839a5650 Driver        LSI_SCSI
         878078b0 Driver        Wanarpv6
         8396d348 Driver        lltdio
     28  87d55030 Driver        VMMEMCTL
         848e20d8 Driver        usbehci
         87746c28 Driver        Null
         877f74a0 Driver        ws2ifsl
     29  83eae3c0 Driver        disk
         83d7f118 Driver        pci
     30  83e53b10 Driver        partmgr
         848ee488 Driver        NdisWan
         87dfd9e0 Driver        NdisTapi
         87dfd030 Driver        Serial
     31  8488a8e8 Driver        DXGKrnl
     32  838c0188 Driver        Wdf01000
         838c1ba8 Driver        ACPI
     33  82b82b08 Driver        PnpManager
         84bfeb88 Driver        flpydisk
     34  8774b3b0 Driver        vmrawdsk
         877f88d0 Driver        AFD
     35  878da110 Driver        Parport
         879ff500 Driver        E1G60
         8776b030 Driver        HidUsb
     36  83934688 Driver        intelide
         87668378 Driver        CmBatt
         84a0c2f0 Driver        i8042prt

1: kd> !object \Device
Object: 8580f2e0  Type: (82b38d60) Directory
    ObjectHeader: 8580f2c8 (old version)
    HandleCount: 0  PointerCount: 256
    Directory Object: 858074c0  Name: Device

    Hash Address  Type          Name
    ---- -------  ----          ----
     00  83eae9d8 Device        KsecDD
         83960668 Device        Ndis
         8598e918 SymbolicLink  ScsiPort2
         87cccd38 Device        SrvNet
         82b41030 Device        00000032
         87746570 Device        Beep
         82b3e458 Device        00000025
         82b3c430 Device        00000019
     01  8776d980 Device        Netbios
         871072c0 SymbolicLink  ScsiPort3
         82b41d80 Device        00000033
         82b3e198 Device        00000026
     02  82b41ad0 Device        00000034
         8825bfe0 SymbolicLink  Ip
         8392a980 Device        00000040
```

```
        82b3fed0 Device        00000027
    03  871ea268 SymbolicLink  {E3FE0F52-6729-43AC-8488-5AC1FB2AE7A9}
        8760e040 Device        Video0
        838c1e38 Device        KeyboardClass0
        82b41850 Device        00000035
        8392a868 Device        00000041
        838c1030 Device        KMDF0
        82b37030 Device        WMIAdminDevice
        82b3fc10 Device        00000028
    04  92b235d0 SymbolicLink  MailslotRedirector
        871dc7d8 SymbolicLink  {6EA11ADB-6FEB-425D-A3CB-3CB73F334E62}
        87747030 Device        Video1
        8760a030 Device        NDProxy
        848e2450 Device        KeyboardClass1
        83930510 Device        VolMgrControl
        8392a750 Device        00000042
        82b41468 Device        00000036
        82b3f950 Device        00000029
    05  848be8a0 Device        Serial0
        87ccb690 Device        SrvAdmin
        877475d8 Device        Video2
        848d1030 Device        PointerClass0
        88240710 SymbolicLink  Ip6
        84b88028 Device        00000050
        8392a638 Device        00000043
        83da6d50 Device        00000037
        82b3adb0 Device        0000000a
    06  84be2258 Device        Video3
        8392d828 Device        00000038
        848de028 Device        USBPDO-0
        848ed648 Device        PointerClass1
        83962778 Device        CompositeBattery
        87665028 Device        00000051
        848a94e0 Device        Serial1
        8392a520 Device        00000044
        82b3ab30 Device        0000000b
    07  87781030 Device        NetBT_Tcpip_{0DC6D9AD-70DC-41CE-9798-F71D1A8C899F}
        839da1e8 Device        SpDevice
        82b37be8 Device        WMIDataDevice
        8760c028 Device        USBPDO-1
        876a6ea0 Device        Video4
        87772328 Device        PointerClass2
        8585ec78 SymbolicLink  {6AF476B1-AA92-4BE1-AA1C-49257F765446}
        8392a408 Device        00000045
        839e6210 Device        00000039
        838a7bf0 Device        RawTape
        82b3a8b0 Device        0000000c
    08  848ebb90 Device        FloppyPDO0
        8760f030 Device        USBPDO-2
        87dad030 Device        PEAuth
        92b1f758 SymbolicLink  WebDavRedirector
        8392a2f0 Device        00000046
        8776f2d0 Device        PointerClass3
        87783030 Device        00000053
        83912098 Device        NTPNP_PCI0000
        82b3c178 Device        0000001a
        82b3a5f8 Device        0000000d
    09  8782e030 Device        USBPDO-3
        87d2d9f8 Device        MPS
        8392b030 Device        00000047
```

```
         8777e030 Device        00000054
         83bab030 Device        NTPNP_PCI0001
         82b3df10 Device        0000001b
         82b3a338 Device        0000000e
[...]
     12  8776f888 Device        00000057
         877bff18 Device        eQoS
         83bc4b98 Device        NTPNP_PCI0011
         82b3f690 Device        0000002a
         82b3d790 Device        0000001e
     13  8452d6c0 Device        HarddiskVolume1
         878ea3d0 Device        NDMP1
         92b12350 Directory     Http
         877e7028 Device        00000058
         82b3f3d0 Device        0000002b
         83bc4700 Device        NTPNP_PCI0012
         83da6030 Device        NTPNP_PCI0005
         82b3d4d8 Device        0000001f
     14  849f3030 Device        CdRom0
         83da68b8 Device        NTPNP_PCI0006
         839d6ab0 Device        ECacheControl
         877f4178 Device        NDMP2
         84be2b38 Device        00000059
         877fc340 Device        FsWrap
         82b40030 Device        0000002c
         848e2a68 Device        Termdd
         83c4b030 Device        NTPNP_PCI0013
     15  859b5d98 Directory     Ide
         8782f030 Device        hgfsInternal
         877f53d0 Device        NDMP3
         877835a8 Device        _HID00000000
         877f6030 Device        RawIp6
         84b1d678 Device        Parallel0
         83babbb0 Device        0000003a
         82b40db0 Device        0000002d
         839ad030 Device        NTPNP_PCI0007
         82b45b98 Device        NTPNP_PCI0020
         83c4bb98 Device        NTPNP_PCI0014
     16  848d0408 Device        NDMP4
         8776bd48 Device        _HID00000001
         82b37180 Device        0000003b
         82b40b30 Device        0000002e
         82b45700 Device        NTPNP_PCI0021
         83c4b700 Device        NTPNP_PCI0015
         839adb28 Device        NTPNP_PCI0008
     17  82b831f0 Event         VolumesSafeForWriteAccess
         848f1400 Device        NDMP5
         82b40870 Device        0000002f
         84a2bec8 Device        vmci
         82b46030 Device        NTPNP_PCI0022
         83cfa030 Device        NTPNP_PCI0016
         839ad690 Device        NTPNP_PCI0009
         8390fda0 Device        0000003c
     18  848e43d0 Device        NDMP6
         87cc9160 Device        Secdrv
         877503a8 Device        Tcp6
         82b7c700 Device        NTPNP_PCI0030
         82b46b98 Device        NTPNP_PCI0023
         83cfab98 Device        NTPNP_PCI0017
         83a51f18 Device        0000003d
```

```
    19  879e43d0 Device        NDMP7
        8776b460 Device        NetBt_Wins_Export
        8392cf18 Device        0000004a
        83913030 Device        NTPNP_PCI0031
        82b46700 Device        NTPNP_PCI0024
        83cfa700 Device        NTPNP_PCI0018
        83a51450 Device        0000003e
    20  877c4e58 Device        WFP
        8392ce00 Device        0000004b
        83a2c030 Device        0000003f
        83913b98 Device        NTPNP_PCI0032
        82b7b030 Device        NTPNP_PCI0025
        82b45030 Device        NTPNP_PCI0019
    21  877e5030 Device        NetbiosSmb
        8392cce8 Device        0000004c
        83913700 Device        NTPNP_PCI0033
        82b7bb98 Device        NTPNP_PCI0026
    22  87da8168 Device        0000005a
        839af6b0 Device        0000004d
        83916b98 Device        NTPNP_PCI0040
        83914030 Device        NTPNP_PCI0034
        82b7b700 Device        NTPNP_PCI0027
    23  83963858 Device        MountPointManager
        879ec730 Device        rspndr
        877d71c8 Device        Tdx
        8392c2d0 Device        NTPNP_PCI0041
        83914b98 Device        NTPNP_PCI0035
        82b7c030 Device        NTPNP_PCI0028
    24  839d5998 Device        RaidPort0
        83e8f7c8 Device        Mup
        87d14098 Device        LanmanServer
        877fce20 Device        Nsi
        87cfd998 Device        Srv2
        8782f798 Device        WANARP
        8392fb98 Device        NTPNP_PCI0042
        8763e030 Device        INTELPRO_{0DC6D9AD-70DC-41CE-9798-F71D1A8C899F}
        848ef030 Device        0000004f
        83914700 Device        NTPNP_PCI0036
        82b7cb98 Device        NTPNP_PCI0029
    25  8392f700 Device        NTPNP_PCI0043
        87115a70 SymbolicLink  {54950694-33A2-408C-9E06-ABBEB791E26F}
        877e6830 Device        Udp
        87900800 Device        RaidPort1
        83915030 Device        NTPNP_PCI0037
    26  87103878 Directory     Harddisk0
        8717ebb8 SymbolicLink  NdisWanIp
        877e6378 Device        RawIp
        83930030 Device        NTPNP_PCI0044
        82b37a58 Device        00000001
        839159c8 Device        NTPNP_PCI0038
    27  87dfddb8 Device        Floppy0
        83978cc8 Device        lltdio
        8782f620 Device        WANARPV6
        838a7e20 Device        RawDisk
        83916030 Device        NTPNP_PCI0039
        82b37738 Device        00000002
    28  848a7028 Device        USBFDO-0
        87d76c30 Device        vmmemctl
        87746b10 Device        Null
        859bb478 SymbolicLink  hgfs
        877f7388 Device        WS2IFSL
        82b3bdb0 Device        00000010
        82b39030 Device        00000003
    29  877ad340 Device        NXTIPSEC
        82b39db0 Device        00000004
        848ab028 Device        USBFDO-1
        82b3baf0 Device        00000011
    30  87da56e0 Device        AscKmd
        87ccbe20 Device        LanmanDatagramReceiver
        85812ef0 Section       PhysicalMemory
        877e6710 Device        Udp6
        8775f030 Device        NdisWan
        87900698 Device        NdisTapi
        82b3b838 Device        00000012
        82b39b30 Device        00000005
    31  92b23470 SymbolicLink  LanmanRedirector
        848c6710 Device        DxgKrnl
        82b3b578 Device        00000013
        82b398b0 Device        00000006
    32  877539e0 Device        NamedPipe
        8599feb8 SymbolicLink  FtControl
        82b3d220 Device        00000020
        82b3b2c0 Device        00000014
        82b39630 Device        00000007
    33  87747d50 Device        Mailslot
        8717ec68 SymbolicLink  NdisWanIpv6
        82b3ef10 Device        00000021
        82b3cf10 Device        00000015
        82b393b0 Device        00000008
    34  877f87b8 Device        Afd
        83959668 Device        FileInfo
        838a7d08 Device        RawCdRom
        82b3ec90 Device        00000022
        82b3cc58 Device        00000016
        82b3a030 Device        00000009
    35  877c6f18 Device        WfpAle
        82b405b0 Device        00000030
        859949a0 SymbolicLink  ScsiPort0
        82b3e9d8 Device        00000023
        82b3c9a0 Device        00000017
    36  82b40300 Device        00000031
        870bf680 SymbolicLink  ScsiPort1
        82b3e718 Device        00000024
        82b3c6e8 Device        00000018
```

Note that if you find any device suspicious, you can get a pointer to its driver object:

```
1: kd> !devobj 877c6f18
[...]

1: kd> !devobj 877c6f18
Device object (877c6f18) is for:
 WfpAle \Driver\Tcpip DriverObject 87745608
Current Irp 00000000 RefCount 1 Type 00000012 Flags 00000040
Dacl 8824c504 DevExt 00000000 DevObjExt 877c6fd0
ExtensionFlags (0000000000)
Characteristics (0x00000100)  FILE_DEVICE_SECURE_OPEN
Device queue is not busy.
```

```
1: kd> dt nt!_DEVICE_OBJECT 877c6f18
ntdll!_DEVICE_OBJECT
   +0x000 Type             : 0n3
   +0x002 Size             : 0xb8
   +0x004 ReferenceCount   : 0n1
   +0x008 DriverObject     : 0x87745608 _DRIVER_OBJECT
   +0x00c NextDevice       : 0x877c4e58 _DEVICE_OBJECT
   +0x010 AttachedDevice   : (null)
   +0x014 CurrentIrp       : (null)
   +0x018 Timer            : (null)
   +0x01c Flags            : 0x40
   +0x020 Characteristics  : 0x100
   +0x024 Vpb              : (null)
   +0x028 DeviceExtension  : (null)
   +0x02c DeviceType       : 0x12
   +0x030 StackSize        : 1 ''
   +0x034 Queue            : <unnamed-tag>
   +0x05c AlignmentRequirement : 0
   +0x060 DeviceQueue      : _KDEVICE_QUEUE
   +0x074 Dpc              : _KDPC
   +0x094 ActiveThreadCount : 0
   +0x098 SecurityDescriptor : 0x8824c4f0 Void
   +0x09c DeviceLock       : _KEVENT
   +0x0ac SectorSize       : 0
   +0x0ae Spare1           : 0
   +0x0b0 DeviceObjectExtension : 0x877c6fd0 _DEVOBJ_EXTENSION
   +0x0b4 Reserved         : (null)

1: kd> !drvobj 0x87745608
Driver object (87745608) is for:
 \Driver\Tcpip
Driver Extension List: (id , addr)

Device Object list:
877bff18  877c6f18  877c4e58  877ad340
877454f0

1: kd> dt nt!_DRIVER_OBJECT 0x87745608
ntdll!_DRIVER_OBJECT
   +0x000 Type             : 0n4
   +0x002 Size             : 0n168
   +0x004 DeviceObject     : 0x877bff18 _DEVICE_OBJECT
   +0x008 Flags            : 0x12
   +0x00c DriverStart      : 0x88b03000 Void
   +0x010 DriverSize       : 0xd1000
   +0x014 DriverSection    : 0x84b1dce8 Void
   +0x018 DriverExtension  : 0x877456b0 _DRIVER_EXTENSION
   +0x01c DriverName       : _UNICODE_STRING "\Driver\Tcpip"
   +0x024 HardwareDatabase : 0x81b02ed8 _UNICODE_STRING
"\REGISTRY\MACHINE\HARDWARE\DESCRIPTION\SYSTEM"
   +0x028 FastIoDispatch   : (null)
   +0x02c DriverInit       : 0x88bc81b9     long  tcpip!GsDriverEntry+0
   +0x030 DriverStartIo    : (null)
   +0x034 DriverUnload     : 0x88bc55b2     void  tcpip!DriverUnload+0
   +0x038 MajorFunction    : [28] 0x88b28e22     long  tcpip!NlDispatchClose+0
```

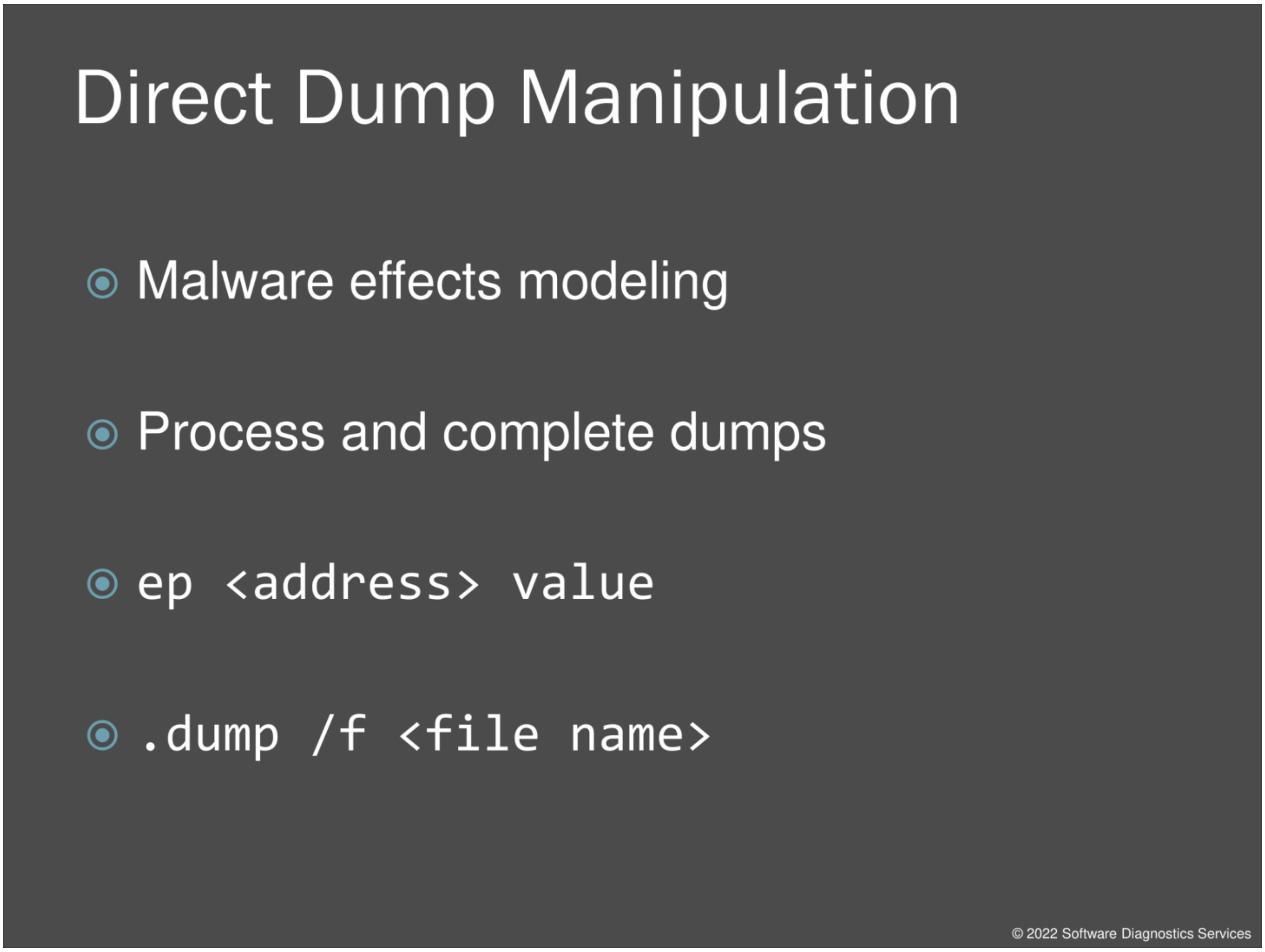
9. Suppose we find a suspicious driver object (for example, from its name or from a problem thread that has an IRP in WinDbg output), then we can check its IRP dispatch table:

```
1: kd> !drvobj \Driver\CmBatt 3
[...]

1: kd> !drvobj \Driver\CmBatt 3
Driver object (87668378) is for:
 \Driver\CmBatt
Driver Extension List: (id , addr)

Device Object list:
849e38a0  848c29b8

DriverEntry:   85a399bc   CmBatt!GsDriverEntry
DriverStartIo: 00000000
DriverUnload:  85a38b06   CmBatt!CmBattUnload
AddDevice:     85a38588   CmBatt!CmBattAddDevice

Dispatch routines:
[00] IRP_MJ_CREATE                      85a38b40    CmBatt!CmBattOpenClose
[01] IRP_MJ_CREATE_NAMED_PIPE           8181d171    nt!IopInvalidDeviceRequest
[02] IRP_MJ_CLOSE                       85a38b40    CmBatt!CmBattOpenClose
[03] IRP_MJ_READ                        87fe6226    E1G60I32!EepromRead
[04] IRP_MJ_WRITE                       8181d171    nt!IopInvalidDeviceRequest
[05] IRP_MJ_QUERY_INFORMATION           8181d171    nt!IopInvalidDeviceRequest
[06] IRP_MJ_SET_INFORMATION             8181d171    nt!IopInvalidDeviceRequest
[07] IRP_MJ_QUERY_EA                    8181d171    nt!IopInvalidDeviceRequest
[08] IRP_MJ_SET_EA                      8181d171    nt!IopInvalidDeviceRequest
[09] IRP_MJ_FLUSH_BUFFERS               8181d171    nt!IopInvalidDeviceRequest
[0a] IRP_MJ_QUERY_VOLUME_INFORMATION    8181d171    nt!IopInvalidDeviceRequest
[0b] IRP_MJ_SET_VOLUME_INFORMATION      8181d171    nt!IopInvalidDeviceRequest
[0c] IRP_MJ_DIRECTORY_CONTROL           8181d171    nt!IopInvalidDeviceRequest
[0d] IRP_MJ_FILE_SYSTEM_CONTROL         8181d171    nt!IopInvalidDeviceRequest
[0e] IRP_MJ_DEVICE_CONTROL              85a38bac    CmBatt!CmBattIoctl
[0f] IRP_MJ_INTERNAL_DEVICE_CONTROL     8181d171    nt!IopInvalidDeviceRequest
[10] IRP_MJ_SHUTDOWN                    8181d171    nt!IopInvalidDeviceRequest
[11] IRP_MJ_LOCK_CONTROL                8181d171    nt!IopInvalidDeviceRequest
[12] IRP_MJ_CLEANUP                     8181d171    nt!IopInvalidDeviceRequest
[13] IRP_MJ_CREATE_MAILSLOT             8181d171    nt!IopInvalidDeviceRequest
[14] IRP_MJ_QUERY_SECURITY              8181d171    nt!IopInvalidDeviceRequest
[15] IRP_MJ_SET_SECURITY                8181d171    nt!IopInvalidDeviceRequest
[16] IRP_MJ_POWER                       85a37ef8    CmBatt!CmBattPowerDispatch
[17] IRP_MJ_SYSTEM_CONTROL              85a39492    CmBatt!CmBattSystemControl
[18] IRP_MJ_DEVICE_CHANGE               8181d171    nt!IopInvalidDeviceRequest
[19] IRP_MJ_QUERY_QUOTA                 8181d171    nt!IopInvalidDeviceRequest
[1a] IRP_MJ_SET_QUOTA                   8181d171    nt!IopInvalidDeviceRequest
[1b] IRP_MJ_PNP                         85a3811c    CmBatt!CmBattPnpDispatch
```

We see that one of the entries (IRP_MJ_READ) points to memory outside of the driver module range.

10. Close the log file:

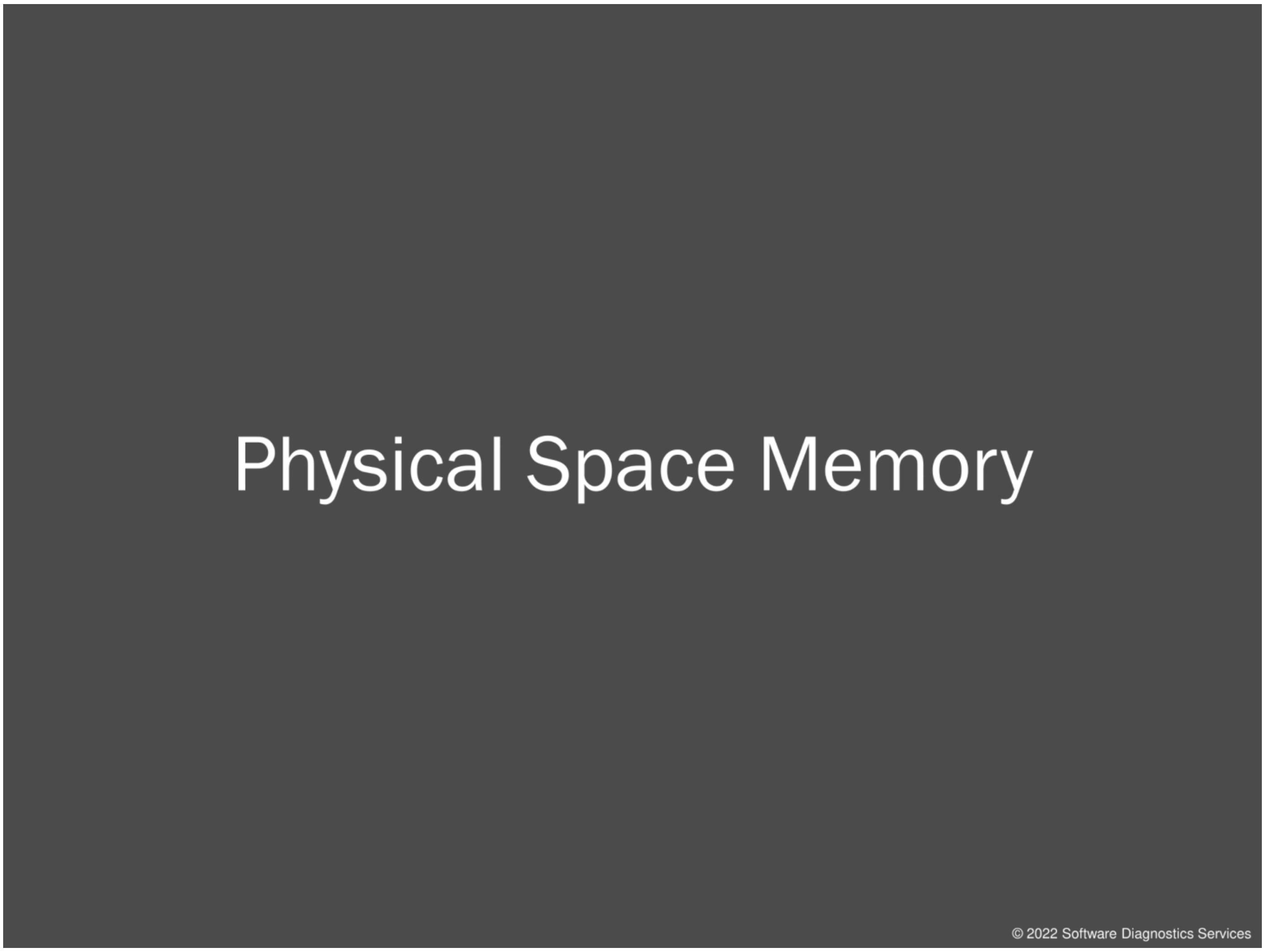
```
1: kd> .logclose
Closing open log file C:\AWMA-Dumps\M5.log
```



Direct Dump Manipulation

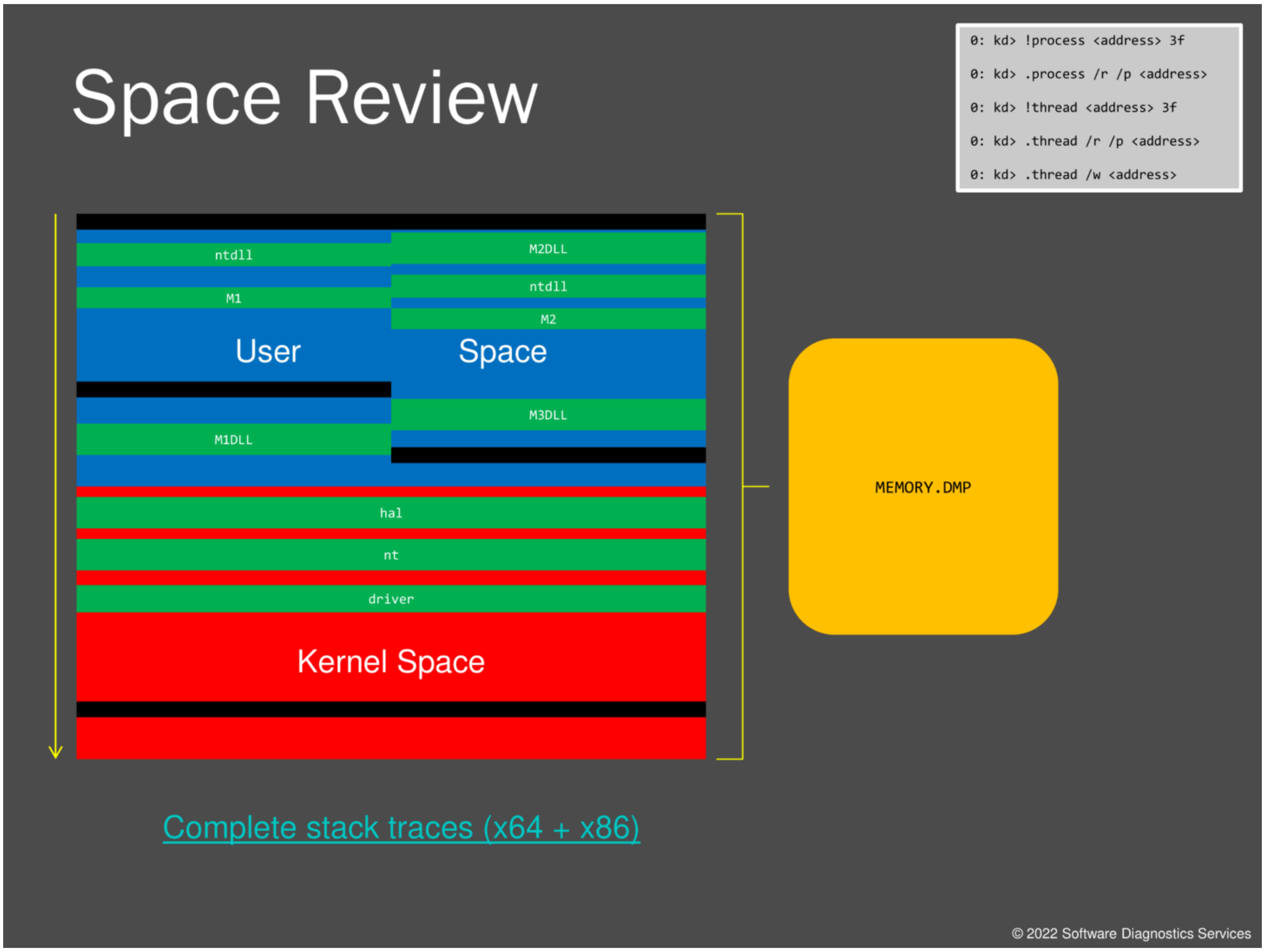
- Malware effects modeling
- Process and complete dumps
- `ep <address> value`
- `.dump /f <file name>`

For this dump, we used the so-called direct dump manipulation (by analogy with a known malware technique called direct kernel object manipulation, DKOM). We just modified some pointers using the **e** command variants such as **ep** and saved a copy using the **.dump** command. Thus we modeled certain malware effects in memory without spending much time writing actual code that does that.



Physical Space Memory

Now we discuss physical memory space. Because we already analyzed a complete memory dump in the M4 exercise, you won't see much difference in our next exercise.



Space Review

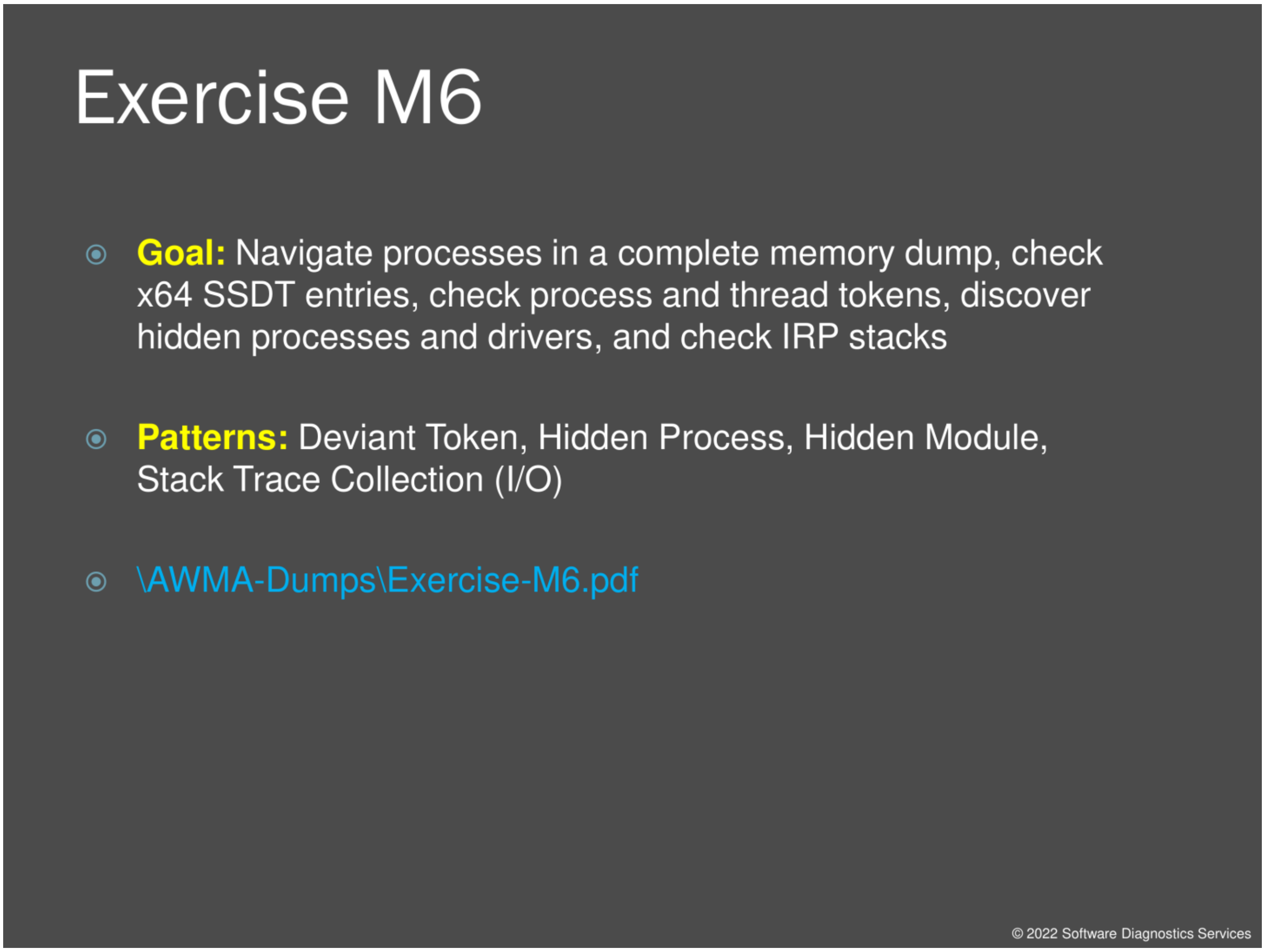
```
0: kd> !process <address> 3f

0: kd> .process /r /p <address>

0: kd> !thread <address> 3f

0: kd> .thread /r /p <address>

0: kd> .thread /w <address>
```

Complete stack traces (x64 + x86)

© 2022 Software Diagnostics Services

In a physical space (and in a complete memory dump), we have several user spaces but only one kernel space. So when we navigate between processes, we need to make sure that we change to the correct user space and reload symbols. Also, for x64 systems, we might have 32-bit processes, and if you use the **!process** command like we did previously, you don't find 32-bit thread stack traces. So for this presentation, I provided a small WinDbg script that dumps both types of stack traces (see also scripts on windbg.org).

Complete stack traces (x64 + x86, also available in Volume 5 of Memory Dump Analysis Anthology and this book Appendix): https://www.dumpanalysis.org/blog/index.php/2010/02/09/complete-stack-traces-from-x64-system/.



Exercise M6

- **Goal:** Navigate processes in a complete memory dump, check x64 SSDT entries, check process and thread tokens, discover hidden processes and drivers, and check IRP stacks

- **Patterns:** Deviant Token, Hidden Process, Hidden Module, Stack Trace Collection (I/O)

- \AWMA-Dumps\Exercise-M6.pdf

Exercise M6

Goal: Navigate processes in a complete memory dump, check x64 SSDT entries, check process and thread tokens, discover hidden processes and drivers, and check IRP stacks.

Patterns: Deviant Token, Hidden Process, Hidden Module, Stack Trace Collection (I/O).

1. Launch WinDbg Preview.

2. Open \AWMA-Dumps\Complete\MEMORY3.DMP.

3. We get the dump file loaded:

```
Microsoft (R) Windows Debugger Version 10.0.25136.1001 AMD64
Copyright (c) Microsoft Corporation. All rights reserved.


Loading Dump File [C:\AWMA-Dumps\Complete\MEMORY3.DMP]
Kernel Bitmap Dump File: Full address space is available


************* Path validation summary **************
Response                         Time (ms)     Location
Deferred                                       srv*
Symbol search path is: srv*
Executable search path is:
Windows 10 Kernel Version 22000 MP (2 procs) Free x64
Product: WinNt, suite: TerminalServer SingleUserTS Personal
Edition build lab: 22000.1.amd64fre.co_release.210604-1628
Machine Name:
Kernel base = 0xfffff807`62000000 PsLoadedModuleList = 0xfffff807`62c29bc0
Debug session time: Thu Feb 10 02:11:26.439 2022 (UTC + 1:00)
System Uptime: 0 days 0:07:45.422
Loading Kernel Symbols
...............................................................
................................................................
................................................................
..
Loading User Symbols
..................................
Loading unloaded module list
........
For analysis of this file, run !analyze -v
nt!KeBugCheckEx:
fffff807`62416220 48894c2408      mov     qword ptr [rsp+8],rcx
ss:0018:ffffa28c`9d8d8690=000000000000000a
```

4. Open a log file:

```
1: kd> .logopen C:\AWMA-Dumps\M6.log
Opened log file 'C:\AWMA-Dumps\M6.log'
```

5. First we check SSDT to see if there is any difference compared to x86 32-bit version:

```
1: kd> dps nt!KeServiceDescriptorTable
fffff807`62e018c0  fffff807`620ca090 nt!KiServiceTable
fffff807`62e018c8  00000000`00000000
fffff807`62e018d0  00000000`000001e1
fffff807`62e018d8  fffff807`620ca818 nt!KiArgumentTable
fffff807`62e018e0  00000000`00000000
fffff807`62e018e8  00000000`00000000
fffff807`62e018f0  00000000`00000000
fffff807`62e018f8  00000000`00000000
fffff807`62e01900  00000000`00000000
fffff807`62e01908  00000000`00000000
fffff807`62e01910  fffff807`62ab22c0 nt!KiBreakpointTrapShadow
fffff807`62e01918  fffff807`62ab2340 nt!KiOverflowTrapShadow
fffff807`62e01920  fffff807`62ab2d40 nt!KiRaiseSecurityCheckFailureShadow
fffff807`62e01928  fffff807`62ab2dc0 nt!KiRaiseAssertionShadow
fffff807`62e01930  fffff807`62ab2e40 nt!KiDebugServiceTrapShadow
fffff807`62e01938  fffff807`62ab4180 nt!KiSystemCall64Shadow
fffff807`62e01940  fffff807`62ab3e40 nt!KiSystemCall32Shadow
```

However, it looks like it is either encrypted or compacted:

```
1: kd> dps nt!KiServiceTable
fffff807`620ca090  016b0c00`01d3f004
fffff807`620ca098  08b18700`05eb0802
fffff807`620ca0a0  034fe600`06a0a900
fffff807`620ca0a8  06a0c506`06b66c05
fffff807`620ca0b0  06b64601`06246505
fffff807`620ca0b8  0681b900`06233900
fffff807`620ca0c0  06a65900`065acf00
fffff807`620ca0c8  06a1eb00`05a5f700
fffff807`620ca0d0  062c8f01`0658bd01
fffff807`620ca0d8  05a8f602`05ac2f00
fffff807`620ca0e0  061df100`06c07e00
fffff807`620ca0e8  068b2202`06b10b01
fffff807`620ca0f0  06b8c101`0618f502
fffff807`620ca0f8  05f55805`05bc8f01
fffff807`620ca100  0650cc03`06165400
fffff807`620ca108  08996a00`06ac2b00
```

Here’s the algorithm for the 4th entry (index 3):

```
; Get the DWORD entry

1: kd> ? dwo(nt!KiServiceTable+4*3)
Evaluate expression: 145852160 = 00000000`08b18700

; if negative sign extend (I haven’t seen negative values in latest Windows versions)
; Example from Windows 8 memory dump:
; 0: kd> ? 00000000`ffff5b00 or ffffffff`00000000
; Evaluate expression: -42240 = ffffffff`ffff5b00

; Right arithmetic shift by 4 bits (sign extended)

1: kd> ? (00000000`08b18700 >>> 4)
Evaluate expression: 9115760 = 00000000`008b1870
```

```
; Add to nt!KiServiceTable address

1: kd> ? nt!KiServiceTable + 00000000`008b1870
Evaluate expression: -8764374140672 = fffff807`6297b900

1: kd> ln fffff807`6297b900
Browse module
Set bu breakpoint

(fffff807`6297b900)   nt!NtMapUserPhysicalPagesScatter   |  (fffff807`6297bc50)
nt!MiBadMemoryLogger
Exact matches:

1: kd> u fffff807`6297b900
nt!NtMapUserPhysicalPagesScatter:
fffff807`6297b900 48895c2420      mov     qword ptr [rsp+20h],rbx
fffff807`6297b905 55              push    rbp
fffff807`6297b906 56              push    rsi
fffff807`6297b907 57              push    rdi
fffff807`6297b908 4154            push    r12
fffff807`6297b90a 4155            push    r13
fffff807`6297b90c 4156            push    r14
fffff807`6297b90e 4157            push    r15
```

6. Now we find Notepad process address from the following explicit command and make it current:

```
1: kd> !process 0 0 Notepad.exe
PROCESS ffffbe0c870210c0
    SessionId: 1  Cid: 1b24    Peb: 4a21ca8000  ParentCid: 1070
    DirBase: 56023002  ObjectTable: ffff800edebca400  HandleCount: 256.
    Image: Notepad.exe

1: kd> .process /r /p ffffbe0c870210c0
Implicit process is now ffffbe0c`870210c0
Loading User Symbols
.....................................................

************* Symbol Loading Error Summary **************
Module name            Error
SharedUserData         No error - symbol load deferred
vmci                   The system cannot find the file specified
myfault                The system cannot find the file specified

You can troubleshoot most symbol related issues by turning on symbol loading diagnostics (!sym
noisy) and repeating the command that caused symbols to be loaded.
You should also verify that your symbol search path (.sympath) is correct.
```

Let's now check its module load address, dump PE header, and check IAT:

```
1: kd> lm m Notepad
Browse full module list
start             end                 module name
00007ff7`b4540000 00007ff7`b4586000   Notepad    (deferred)

1: kd> !dh 00007ff7`b4540000

File Type: EXECUTABLE IMAGE
FILE HEADER VALUES
    8664 machine (X64)
       6 number of sections
```

```
60622CE6 time date stamp Mon Mar 29 20:39:18 2021

       0 file pointer to symbol table
       0 number of symbols
      F0 size of optional header
      22 characteristics
            Executable
            App can handle >2gb addresses

OPTIONAL HEADER VALUES
     20B magic #
   14.28 linker version
   21A00 size of code
   20000 size of initialized data
       0 size of uninitialized data
   20D84 address of entry point
    1000 base of code
         ----- new -----
00007ff7b4540000 image base
    1000 section alignment
     200 file alignment
       2 subsystem (Windows GUI)
    6.00 operating system version
    0.00 image version
    6.00 subsystem version
   46000 size of image
     400 size of headers
       0 checksum
0000000000100000 size of stack reserve
0000000000001000 size of stack commit
0000000000100000 size of heap reserve
0000000000001000 size of heap commit
    8160  DLL characteristics
            High entropy VA supported
            Dynamic base
            NX compatible
            Terminal server aware
       0 [       0] address [size] of Export Directory
   2D890 [     244] address [size] of Import Directory
   37000 [    DA00] address [size] of Resource Directory
   35000 [    1110] address [size] of Exception Directory
       0 [       0] address [size] of Security Directory
   45000 [     38C] address [size] of Base Relocation Directory
   27488 [      70] address [size] of Debug Directory
       0 [       0] address [size] of Description Directory
       0 [       0] address [size] of Special Directory
   27680 [      28] address [size] of Thread Storage Directory
   27500 [     138] address [size] of Load Configuration Directory
       0 [       0] address [size] of Bound Import Directory
   23000 [     A50] address [size] of Import Address Table Directory
       0 [       0] address [size] of Delay Import Directory
       0 [       0] address [size] of COR20 Header Directory
       0 [       0] address [size] of Reserved Directory


SECTION HEADER #1
   .text name
   21817 virtual size
    1000 virtual address
   21A00 size of raw data
```

```
     400 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
60000020 flags
         Code
         (no align specified)
         Execute Read

SECTION HEADER #2
  .rdata name
    CDB6 virtual size
   23000 virtual address
    CE00 size of raw data
   21E00 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
40000040 flags
         Initialized Data
         (no align specified)
         Read Only


Debug Directories(4)
       Type       Size     Address  Pointer
       cv           41       2a174    28f74   Format: RSDS, guid, 1,
D:\a\1\b\Release\x64\Notepad\Notepad.pdb
       (   12)      14       2a1b8    28fb8
       (   13)     3e8       2a1cc    28fcc
       (   14)       0           0        0

SECTION HEADER #3
   .data name
    40E0 virtual size
   30000 virtual address
    2A00 size of raw data
   2EC00 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
C0000040 flags
         Initialized Data
         (no align specified)
         Read Write

SECTION HEADER #4
  .pdata name
    1110 virtual size
   35000 virtual address
    1200 size of raw data
   31600 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
40000040 flags
```

```
         Initialized Data
         (no align specified)
         Read Only

SECTION HEADER #5
   .rsrc name
    DA00 virtual size
   37000 virtual address
    DA00 size of raw data
   32800 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
40000040 flags
         Initialized Data
         (no align specified)
         Read Only

SECTION HEADER #6
  .reloc name
     38C virtual size
   45000 virtual address
     400 size of raw data
   40200 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
42000040 flags
         Initialized Data
         Discardable
         (no align specified)
         Read Only
```

```
1: kd> dps 00007ff7b4540000+23000 LA50/8
00007ff7`b4563000  00007ffe`5a216b20 ADVAPI32!RegCloseKeyStub
00007ff7`b4563008  00007ffe`5a23d090 ADVAPI32!DuplicateEncryptionInfoFile
00007ff7`b4563010  00007ffe`5a217680 ADVAPI32!RegCreateKeyExWStub
00007ff7`b4563018  00007ffe`5a216750 ADVAPI32!RegQueryValueExWStub
00007ff7`b4563020  00007ffe`5a218460 ADVAPI32!RegCreateKeyW
00007ff7`b4563028  00007ffe`5af865e0 ntdll!EtwEventUnregister
00007ff7`b4563030  00007ffe`5a22f950 ADVAPI32!RegDeleteKeyExWStub
00007ff7`b4563038  00007ffe`5a2168e0 ADVAPI32!GetTokenInformationStub
00007ff7`b4563040  00007ffe`5a216b40 ADVAPI32!IsTextUnicode
00007ff7`b4563048  00007ffe`5af84f40 ntdll!EtwEventWriteTransfer
00007ff7`b4563050  00007ffe`5a216900 ADVAPI32!RegQueryInfoKeyWStub
00007ff7`b4563058  00007ffe`5a216d50 ADVAPI32!RegEnumValueWStub
00007ff7`b4563060  00007ffe`5af95520 ntdll!EtwEventSetInformation
00007ff7`b4563068  00007ffe`5a23d000 ADVAPI32!DecryptFileW
00007ff7`b4563070  00007ffe`5a216800 ADVAPI32!RegOpenKeyExWStub
00007ff7`b4563078  00007ffe`5af959f0 ntdll!EtwEventRegister
00007ff7`b4563080  00007ffe`5a218170 ADVAPI32!RegSetValueExWStub
00007ff7`b4563088  00000000`00000000
00007ff7`b4563090  00007ffe`4445feb0 COMCTL32!TaskDialogIndirect
00007ff7`b4563098  00007ffe`4447a3e0 COMCTL32!CreateStatusWindowW
00007ff7`b45630a0  00000000`00000000
00007ff7`b45630a8  00007ffe`5aaedb70 COMDLG32!ChooseFontW
00007ff7`b45630b0  00007ffe`5aaec7f0 COMDLG32!FindTextW
00007ff7`b45630b8  00007ffe`5aa91440 COMDLG32!GetFileTitleW
```

```
00007ff7`b45630c0  00007ffe`5aaecef0 COMDLG32!ReplaceTextW
00007ff7`b45630c8  00007ffe`5aae6bd0 COMDLG32!GetSaveFileNameW
00007ff7`b45630d0  00007ffe`5aaf32d0 COMDLG32!PageSetupDlgW
00007ff7`b45630d8  00007ffe`5aae6ad0 COMDLG32!GetOpenFileNameW
00007ff7`b45630e0  00007ffe`5aae1b80 COMDLG32!CommDlgExtendedError
00007ff7`b45630e8  00007ffe`5ab215c0 COMDLG32!PrintDlgExW
00007ff7`b45630f0  00000000`00000000
00007ff7`b45630f8  00007ffe`58e25a80 GDI32!EndPage
00007ff7`b4563100  00007ffe`58e27610 GDI32!TextOutW
00007ff7`b4563108  00007ffe`58e2a9d0 GDI32!SetAbortProc
00007ff7`b4563110  00007ffe`58e2e290 GDI32!StartDocW
00007ff7`b4563118  00007ffe`58e23d60 GDI32!SetBkMode
00007ff7`b4563120  00007ffe`58e25cf0 GDI32!EndDoc
00007ff7`b4563128  00007ffe`58e245c0 GDI32!LPtoDPStub
00007ff7`b4563130  00007ffe`58e2c620 GDI32!SetWindowExtExStub
00007ff7`b4563138  00007ffe`58e212d0 GDI32!GetTextExtentPoint32WStub
00007ff7`b4563140  00007ffe`58e2c5b0 GDI32!SetViewportExtExStub
00007ff7`b4563148  00007ffe`58e2d2b0 GDI32!AbortDoc
00007ff7`b4563150  00007ffe`58e273b0 GDI32!EnumFontsW
00007ff7`b4563158  00007ffe`58e275b0 GDI32!GetTextFaceW
00007ff7`b4563160  00007ffe`58e22ef0 GDI32!DeleteDC
00007ff7`b4563168  00007ffe`58e213a0 GDI32!CreateDCW
00007ff7`b4563170  00007ffe`58e25a30 GDI32!StartPage
00007ff7`b4563178  00007ffe`58e23fb0 GDI32!GetTextMetricsWStub
00007ff7`b4563180  00007ffe`58e233d0 GDI32!GetDeviceCaps
00007ff7`b4563188  00007ffe`58e23a90 GDI32!SelectObject
00007ff7`b4563190  00007ffe`58e216e0 GDI32!SetMapModeStub
00007ff7`b4563198  00007ffe`58e21350 GDI32!CreateFontIndirectW
00007ff7`b45631a0  00007ffe`58e21c70 GDI32!DeleteObject
00007ff7`b45631a8  00000000`00000000
00007ff7`b45631b0  00007ffe`5a2e33a0 KERNEL32!MulDiv
00007ff7`b45631b8  00007ffe`5a2db7d0 KERNEL32!IsProcessorFeaturePresentStub
00007ff7`b45631c0  00007ffe`5a2df800 KERNEL32!TerminateProcessStub
00007ff7`b45631c8  00007ffe`5a2e2bd0 KERNEL32!GetCurrentProcess
00007ff7`b45631d0  00007ffe`5afbb4d0 ntdll!RtlLeaveCriticalSection
00007ff7`b45631d8  00007ffe`5afba4e0 ntdll!RtlEnterCriticalSection
00007ff7`b45631e0  00007ffe`5a2de6d0 KERNEL32!SetUnhandledExceptionFilterStub
00007ff7`b45631e8  00007ffe`5af86c60 ntdll!RtlInterlockedPushEntrySList
00007ff7`b45631f0  00007ffe`5a2e3270 KERNEL32!ReadFile
00007ff7`b45631f8  00007ffe`5a2e2d50 KERNEL32!InitializeCriticalSectionAndSpinCount
00007ff7`b4563200  00007ffe`5af9e080 ntdll!RtlDeleteCriticalSection
00007ff7`b4563208  00007ffe`5a2d6030 KERNEL32!GlobalUnlock
00007ff7`b4563210  00007ffe`5a2f8630 KERNEL32!DebugBreakStub
00007ff7`b4563218  00007ffe`5a2db790 KERNEL32!GetModuleHandleWStub
00007ff7`b4563220  00007ffe`5a2dd660 KERNEL32!GetCommandLineWStub
00007ff7`b4563228  00007ffe`5a2de9c0 KERNEL32!HeapSetInformationStub
00007ff7`b4563230  00007ffe`5aff55b0 ntdll!RtlInitializeSListHead
00007ff7`b4563238  00007ffe`5a2d6340 KERNEL32!GetProcessHeapStub
00007ff7`b4563240  00007ffe`5a2e2be0 KERNEL32!GetCurrentProcessId
00007ff7`b4563248  00007ffe`5a2e2d00 KERNEL32!CreateMutexExW
00007ff7`b4563250  00007ffe`5a2fa370 KERNEL32!UnhandledExceptionFilterStub
00007ff7`b4563258  00007ffe`5afa8ac0 ntdll!RtlAllocateHeap
00007ff7`b4563260  00007ffe`5a2e2da0 KERNEL32!OpenSemaphoreW
00007ff7`b4563268  00007ffe`5a2e2e50 KERNEL32!WaitForSingleObjectEx
00007ff7`b4563270  00007ffe`5a2d7a90 KERNEL32!GetSystemTimeAsFileTimeStub
00007ff7`b4563278  00007ffe`5a2e2dc0 KERNEL32!ReleaseMutex
00007ff7`b4563280  00007ffe`5a2e2e40 KERNEL32!WaitForSingleObject
00007ff7`b4563288  00007ffe`5a2de090 KERNEL32!GetModuleHandleExWStub
00007ff7`b4563290  00007ffe`5a2e2dd0 KERNEL32!ReleaseSemaphore
00007ff7`b4563298  00007ffe`5a2d5ef0 KERNEL32!HeapFreeStub
```

```
00007ff7`b45632a0  00007ffe`5a2e2d20 KERNEL32!CreateSemaphoreExW
00007ff7`b45632a8  00007ffe`5a2dd600 KERNEL32!GetModuleFileNameAStub
00007ff7`b45632b0  00007ffe`5a2e5860 KERNEL32!FoldStringWStub
00007ff7`b45632b8  00007ffe`5a2dedf0 KERNEL32!GetLocaleInfoWStub
00007ff7`b45632c0  00007ffe`5a2d6b30 KERNEL32!GlobalFreeStub
00007ff7`b45632c8  00007ffe`5a2e2f30 KERNEL32!FindClose
00007ff7`b45632d0  00007ffe`5a2e2c50 KERNEL32!CloseHandle
00007ff7`b45632d8  00007ffe`5a2dbfa0 KERNEL32!GetModuleFileNameWStub
00007ff7`b45632e0  00007ffe`5a2e2fb0 KERNEL32!FindFirstFileW
00007ff7`b45632e8  00007ffe`5a2df9e0 KERNEL32!GetUserDefaultUILanguageStub
00007ff7`b45632f0  00007ffe`5a2de680 KERNEL32!GetLocalTimeStub
00007ff7`b45632f8  00007ffe`5a2df4a0 KERNEL32!GetDateFormatWStub
00007ff7`b4563300  00007ffe`5a2dd6a0 KERNEL32!GetTimeFormatWStub
00007ff7`b4563308  00007ffe`5b013740 ntdll!_C_specific_handler
00007ff7`b4563310  00007ffe`5a2e2f00 KERNEL32!DeleteFileW
00007ff7`b4563318  00007ffe`5a2d6010 KERNEL32!WideCharToMultiByteStub
00007ff7`b4563320  00007ffe`5a2e3360 KERNEL32!WriteFile
00007ff7`b4563328  00007ffe`5a2e30d0 KERNEL32!GetFileAttributesW
00007ff7`b4563330  00007ffe`5a2e59d0 KERNEL32!LocalLockStub
00007ff7`b4563338  00007ffe`5a2de050 KERNEL32!GetACPStub
00007ff7`b4563340  00007ffe`5a2e59f0 KERNEL32!LocalUnlockStub
00007ff7`b4563348  00007ffe`5a2e32c0 KERNEL32!SetEndOfFile
00007ff7`b4563350  00007ffe`5a2e30c0 KERNEL32!GetFileAttributesExW
00007ff7`b4563358  00007ffe`5a2d6670 KERNEL32!QueryPerformanceCounterStub
00007ff7`b4563360  00007ffe`5a2d5fc0 KERNEL32!MultiByteToWideCharStub
00007ff7`b4563368  00007ffe`5a2de8e0 KERNEL32!LocalReAllocStub
00007ff7`b4563370  00007ffe`5a2dc070 KERNEL32!UnmapViewOfFileStub
00007ff7`b4563378  00007ffe`5a2e30e0 KERNEL32!GetFileInformationByHandle
00007ff7`b4563380  00007ffe`5a2da7c0 KERNEL32!CreateFileMappingWStub
00007ff7`b4563388  00007ffe`5a2db9e0 KERNEL32!MapViewOfFileStub
00007ff7`b4563390  00007ffe`5a2d9330 KERNEL32!LocalAllocStub
00007ff7`b4563398  00007ffe`5a2e2ed0 KERNEL32!CreateFileW
00007ff7`b45633a0  00007ffe`58a1be90 KERNELBASE!GetCurrentPackageFullName
00007ff7`b45633a8  00007ffe`5a2d9370 KERNEL32!GlobalAllocStub
00007ff7`b45633b0  00007ffe`5a2e3160 KERNEL32!GetFullPathNameW
00007ff7`b45633b8  00007ffe`58a6fc20 KERNELBASE!ParseApplicationUserModelId
00007ff7`b45633c0  00007ffe`58a0c250 KERNELBASE!GetCurrentApplicationUserModelId
00007ff7`b45633c8  00007ffe`5a2e2ea0 KERNEL32!CreateDirectoryW
00007ff7`b45633d0  00007ffe`5a2d6360 KERNEL32!SetLastErrorStub
00007ff7`b45633d8  00007ffe`5a2e5ab0 KERNEL32!RtlVirtualUnwindStub
00007ff7`b45633e0  00007ffe`5a2e0b20 KERNEL32!RtlLookupFunctionEntryStub
00007ff7`b45633e8  00007ffe`5a2e2a00 KERNEL32!RtlCaptureContext
00007ff7`b45633f0  00007ffe`5a2e2df0 KERNEL32!SetEvent
00007ff7`b45633f8  00007ffe`5a2e2cc0 KERNEL32!CreateEventExW
00007ff7`b4563400  00007ffe`5a2d7b20 KERNEL32!CompareStringOrdinalStub
00007ff7`b4563408  00007ffe`5a2de920 KERNEL32!GetCurrentDirectoryWStub
00007ff7`b4563410  00007ffe`5a2e0230 KERNEL32!RegisterApplicationRestartStub
00007ff7`b4563418  00007ffe`5a2dba00 KERNEL32!GetStartupInfoWStub
00007ff7`b4563420  00007ffe`5a2e2de0 KERNEL32!ResetEvent
00007ff7`b4563428  00007ffe`5a2e3060 KERNEL32!GetDiskFreeSpaceExW
00007ff7`b4563430  00007ffe`5a2e2cd0 KERNEL32!CreateEventW
00007ff7`b4563438  00007ffe`5a2e01a0 KERNEL32!SetCurrentDirectoryWStub
00007ff7`b4563440  00007ffe`5a2d62e0 KERNEL32!GetLastErrorStub
00007ff7`b4563448  00007ffe`5a2de730 KERNEL32!IsDebuggerPresentStub
00007ff7`b4563450  00007ffe`5a2e5840 KERNEL32!FindNLSStringStub
00007ff7`b4563458  00007ffe`5a2dbf80 KERNEL32!FormatMessageWStub
00007ff7`b4563460  00007ffe`5a2c6170 KERNEL32!GetCurrentThreadId
00007ff7`b4563468  00007ffe`5a2d7b00 KERNEL32!lstrcmpiWStub
00007ff7`b4563470  00007ffe`5a2daf10 KERNEL32!OutputDebugStringWStub
00007ff7`b4563478  00007ffe`5a2d82d0 KERNEL32!LocalFreeStub
```

```
00007ff7`b4563480  00007ffe`5a2d6100 KERNEL32!GlobalLock
00007ff7`b4563488  00007ffe`5a2d93b0 KERNEL32!GetProcAddressStub
00007ff7`b4563490  00007ffe`5a2dbe40 KERNEL32!RaiseExceptionStub
00007ff7`b4563498  00000000`00000000
00007ff7`b45634a0  00007ffe`4658b140 MSVCP140!std::_Xlength_error
[d:\a01\_work\3\s\src\vctools\crt\github\stl\src\xthrow.cpp @ 20]
00007ff7`b45634a8  00000000`00000000
00007ff7`b45634b0  00007ffe`5a54a940 OLEAUT32!SysFreeString
00007ff7`b45634b8  00000000`00000000
00007ff7`b45634c0  00007ffe`561d4820 PROPSYS!PropVariantToStringVectorAlloc
00007ff7`b45634c8  00007ffe`561ba310 PROPSYS!PSGetPropertyDescriptionListFromString
00007ff7`b45634d0  00000000`00000000
00007ff7`b45634d8  00007ffe`599e51b0 SHELL32!ShellExecuteW
00007ff7`b45634e0  00007ffe`599b8900 SHELL32!DragQueryFileW
00007ff7`b45634e8  00007ffe`59884610 SHELL32!SHCreateItemFromParsingName
00007ff7`b45634f0  00007ffe`598e9380 SHELL32!SHAddToRecentDocs
00007ff7`b45634f8  00007ffe`5982f0e0 SHELL32!DragAcceptFiles
00007ff7`b4563500  00007ffe`599b85e0 SHELL32!DragFinish
00007ff7`b4563508  00007ffe`598c3a20 SHELL32!SHGetKnownFolderPathStub
00007ff7`b4563510  00000000`00000000
00007ff7`b4563518  00007ffe`5adb8b30 SHLWAPI!PathIsNetworkPathWStub
00007ff7`b4563520  00007ffe`5adb8d30 SHLWAPI!PathFileExistsWStub
00007ff7`b4563528  00007ffe`5adb16f0 SHLWAPI!SHStrDupWStub
00007ff7`b4563530  00007ffe`5adb7eb0 SHLWAPI!PathFindExtensionWStub
00007ff7`b4563538  00007ffe`5adc1280 SHLWAPI!PathIsFileSpecWStub
00007ff7`b4563540  00000000`00000000
00007ff7`b4563548  00007ffe`5901bba0 USER32!DrawTextExW
00007ff7`b4563550  00007ffe`5902b1a0 USER32!EnableWindow
00007ff7`b4563558  00007ffe`5900dfb0 USER32!GetWindowTextW
00007ff7`b4563560  00007ffe`59019d40 USER32!PeekMessageW
00007ff7`b4563568  00007ffe`590125f0 USER32!GetWindowLongW
00007ff7`b4563570  00007ffe`59013d70 USER32!GetWindowTextLengthW
00007ff7`b4563578  00007ffe`59007bb0 USER32!RegisterClassExW
00007ff7`b4563580  00007ffe`59009080 USER32!LoadImageW
00007ff7`b4563588  00007ffe`59009760 USER32!LoadIconW
00007ff7`b4563590  00007ffe`5908c180 USER32!SetProcessDefaultLayout
00007ff7`b4563598  00007ffe`5900b110 USER32!LoadCursorW
00007ff7`b45635a0  00007ffe`59027130 USER32!RegisterWindowMessageW
00007ff7`b45635a8  00007ffe`590246a0 USER32!MonitorFromWindow
00007ff7`b45635b0  00007ffe`59008030 USER32!CreateWindowExW
00007ff7`b45635b8  00007ffe`5900d440 USER32!SetWindowLongW
00007ff7`b45635c0  00007ffe`590325c0 USER32!NtUserGetSystemMenu
00007ff7`b45635c8  00007ffe`59025f30 USER32!CharUpperWStub
00007ff7`b45635d0  00007ffe`590330c0 USER32!NtUserSetWindowPlacement
00007ff7`b45635d8  00007ffe`590326a0 USER32!NtUserGetWindowPlacement
00007ff7`b45635e0  00007ffe`590309c0 USER32!CreateMenu
00007ff7`b45635e8  00007ffe`59031f30 USER32!NtUserCreateAcceleratorTable
00007ff7`b45635f0  00007ffe`59029940 USER32!UpdateWindow
00007ff7`b45635f8  00007ffe`59032810 USER32!NtUserInvalidateRect
00007ff7`b4563600  00007ffe`5907fd30 USER32!SetScrollPos
00007ff7`b4563608  00007ffe`59019a20 USER32!GetParent
00007ff7`b4563610  00007ffe`5900cf10 USER32!GetCurrentThreadDesktopHwnd
00007ff7`b4563618  00007ffe`590103b0 USER32!GetWindowRect
00007ff7`b4563620  00007ffe`59033200 USER32!NtUserUnhookWinEvent
00007ff7`b4563628  00007ffe`590013f0 USER32!SendDlgItemMessageW
00007ff7`b4563630  00007ffe`5908cb90 USER32!GetDlgItemTextW
00007ff7`b4563638  00007ffe`59029160 USER32!CheckMenuItem
00007ff7`b4563640  00007ffe`5902c540 USER32!CloseClipboardStub
00007ff7`b4563648  00007ffe`590250a0 USER32!IsClipboardFormatAvailableStub
00007ff7`b4563650  00007ffe`5902bb80 USER32!OpenClipboard
```

```
00007ff7`b4563658  00007ffe`59029ef0 USER32!GetSubMenu
00007ff7`b4563660  00007ffe`5902b350 USER32!GetMenu
00007ff7`b4563668  00007ffe`59010bf0 USER32!DispatchMessageW
00007ff7`b4563670  00007ffe`590163e0 USER32!TranslateMessage
00007ff7`b4563678  00007ffe`59015f80 USER32!IsDialogMessageW
00007ff7`b4563680  00007ffe`59024ea0 USER32!TranslateAcceleratorW
00007ff7`b4563688  00007ffe`59024620 USER32!GetMessageW
00007ff7`b4563690  00007ffe`59028a70 USER32!SetWinEventHook
00007ff7`b4563698  00007ffe`5902b130 USER32!CharNextWStub
00007ff7`b45636a0  00007ffe`590277b0 USER32!GetKeyboardLayout
00007ff7`b45636a8  00007ffe`59032be0 USER32!NtUserRedrawWindow
00007ff7`b45636b0  00007ffe`590330d0 USER32!NtUserSetWindowPos
00007ff7`b45636b8  00007ffe`59032310 USER32!NtUserGetForegroundWindow
00007ff7`b45636c0  00007ffe`5908bbf0 USER32!MessageBeep
00007ff7`b45636c8  00007ffe`59031fe0 USER32!NtUserDestroyWindow
00007ff7`b45636d0  00007ffe`5902b7b0 USER32!PostQuitMessage
00007ff7`b45636d8  00007ffe`59024990 USER32!IsIconic
00007ff7`b45636e0  00007ffe`5b023420 ntdll!NtdllDefWindowProc_W
00007ff7`b45636e8  00007ffe`590290c0 USER32!EnableMenuItem
00007ff7`b45636f0  00007ffe`59032d50 USER32!NtUserSetActiveWindow
00007ff7`b45636f8  00007ffe`5902a010 USER32!SetCursorStub
00007ff7`b4563700  00007ffe`59012e70 USER32!GetDpiForWindow
00007ff7`b4563708  00007ffe`5900af60 USER32!ReleaseDC
00007ff7`b4563710  00007ffe`59026d60 USER32!GetDC
00007ff7`b4563718  00007ffe`59033150 USER32!NtUserShowWindow
00007ff7`b4563720  00007ffe`5907cab0 USER32!MessageBoxW
00007ff7`b4563728  00007ffe`59019480 USER32!GetFocus
00007ff7`b4563730  00007ffe`59017070 USER32!PostMessageW
00007ff7`b4563738  00007ffe`59015160 USER32!SetThreadDpiAwarenessContext
00007ff7`b4563740  00007ffe`59010600 USER32!SendMessageW
00007ff7`b4563748  00007ffe`59032980 USER32!NtUserMoveWindow
00007ff7`b4563750  00007ffe`59016b90 USER32!GetClientRect
00007ff7`b4563758  00007ffe`5904fbd0 USER32!DialogBoxParamW
00007ff7`b4563760  00007ffe`59058bf0 USER32!EndDialog
00007ff7`b4563768  00007ffe`59002920 USER32!GetDlgItem
00007ff7`b4563770  00007ffe`59032ec0 USER32!NtUserSetFocus
00007ff7`b4563778  00007ffe`5901ce00 USER32!GetDlgCtrlID
00007ff7`b4563780  00007ffe`5908cc30 USER32!SetDlgItemTextW
00007ff7`b4563788  00007ffe`59015be0 USER32!SetWindowTextW
00007ff7`b4563790  00007ffe`59002f60 USER32!CreateDialogParamW
00007ff7`b4563798  00007ffe`5901aea0 USER32!AppendMenuW
00007ff7`b45637a0  00000000`00000000
00007ff7`b45637a8  00007ffe`465212f0 VCRUNTIME140!memcpy
[d:\a01\_work\3\s\src\vctools\crt\vcruntime\src\string\amd64\memcpy.asm @ 68]
00007ff7`b45637b0  00007ffe`46522540 VCRUNTIME140!__std_terminate
[d:\a01\_work\3\s\src\vctools\crt\vcruntime\src\eh\ehhelpers.cpp @ 191]
00007ff7`b45637b8  00007ffe`46526190 VCRUNTIME140!__std_exception_copy
[d:\a01\_work\3\s\src\vctools\crt\vcruntime\src\eh\std_exception.cpp @ 17]
00007ff7`b45637c0  00007ffe`46526220 VCRUNTIME140!__std_exception_destroy
[d:\a01\_work\3\s\src\vctools\crt\vcruntime\src\eh\std_exception.cpp @ 43]
00007ff7`b45637c8  00007ffe`46526c30 VCRUNTIME140!_purecall
[d:\a01\_work\3\s\src\vctools\crt\vcruntime\src\misc\purevirt.cpp @ 19]
00007ff7`b45637d0  00007ffe`46521fd0 VCRUNTIME140!wcschr
[d:\a01\_work\3\s\src\vctools\crt\vcruntime\src\string\amd64\wcschr.c @ 48]
00007ff7`b45637d8  00007ffe`465224e0 VCRUNTIME140!__current_exception
[d:\a01\_work\3\s\src\vctools\crt\vcruntime\src\eh\ehhelpers.cpp @ 114]
00007ff7`b45637e0  00007ffe`46522500 VCRUNTIME140!__current_exception_context
[d:\a01\_work\3\s\src\vctools\crt\vcruntime\src\eh\ehhelpers.cpp @ 119]
00007ff7`b45637e8  00007ffe`46526430 VCRUNTIME140!_CxxThrowException
[d:\a01\_work\3\s\src\vctools\crt\vcruntime\src\eh\throw.cpp @ 30]
```

```
00007ff7`b45637f0  00007ffe`465219a0 VCRUNTIME140!memset
[d:\a01\_work\3\s\src\vctools\crt\vcruntime\src\string\amd64\memset.asm @ 79]
00007ff7`b45637f8  00007ffe`465212f0 VCRUNTIME140!memcpy
[d:\a01\_work\3\s\src\vctools\crt\vcruntime\src\string\amd64\memcpy.asm @ 68]
00007ff7`b4563800  00000000`00000000
00007ff7`b4563808  00007ffe`46544070 VCRUNTIME140_1!__CxxFrameHandler4
[d:\a01\_work\3\s\src\vctools\crt\vcruntime\src\eh\risctrnsctrl.cpp @ 291]
00007ff7`b4563810  00000000`00000000
00007ff7`b4563818  00007ffe`3aa267d0 WINSPOOL!GetPrinterDriverW
00007ff7`b4563820  00007ffe`3aa31420 WINSPOOL!OpenPrinterW
00007ff7`b4563828  00007ffe`3aa25bc0 WINSPOOL!ClosePrinter
00007ff7`b4563830  00000000`00000000
00007ff7`b4563838  00007ffe`5a78c590 combase!GetRestrictedErrorInfo
[onecore\com\combase\winrt\error\restrictederror.cpp @ 161]
00007ff7`b4563840  00000000`00000000
00007ff7`b4563848  00007ffe`5a7c0f10 combase!RoOriginateLanguageException
[onecore\com\combase\winrt\error\error.cpp @ 1506]
00007ff7`b4563850  00000000`00000000
00007ff7`b4563858  00007ffe`5a746520 combase!RoGetActivationFactory
[onecore\com\combase\winrtbase\winrtbase.cpp @ 1060]
00007ff7`b4563860  00000000`00000000
00007ff7`b4563868  00007ffe`5a717280 combase!WindowsGetStringRawBuffer
[onecore\com\combase\winrt\string\string.cpp @ 226]
00007ff7`b4563870  00007ffe`5a710ac0 combase!WindowsCreateStringReference
[onecore\com\combase\winrt\string\string.cpp @ 70]
00007ff7`b4563878  00007ffe`5a713870 combase!WindowsCreateString
[onecore\com\combase\winrt\string\string.cpp @ 30]
00007ff7`b4563880  00007ffe`5a7400b0 combase!WindowsDeleteString
[onecore\com\combase\winrt\string\string.cpp @ 146]
00007ff7`b4563888  00007ffe`5a78e250 combase!WindowsGetStringLen
[onecore\com\combase\winrt\string\string.cpp @ 202]
00007ff7`b4563890  00000000`00000000
00007ff7`b4563898  00007ffe`584fe1b0 ucrtbase!wtol
00007ff7`b45638a0  00000000`00000000
00007ff7`b45638a8  00007ffe`58502150 ucrtbase!free
00007ff7`b45638b0  00007ffe`58516ae0 ucrtbase!_set_new_mode
00007ff7`b45638b8  00007ffe`58500060 ucrtbase!malloc
00007ff7`b45638c0  00007ffe`58568870 ucrtbase!callnewh
00007ff7`b45638c8  00000000`00000000
00007ff7`b45638d0  00007ffe`58516900 ucrtbase!_configthreadlocale
00007ff7`b45638d8  00000000`00000000
00007ff7`b45638e0  00007ffe`58590d20 ucrtbase!_setusermatherr
00007ff7`b45638e8  00000000`00000000
00007ff7`b45638f0  00007ffe`58515c00 ucrtbase!crt_atexit
00007ff7`b45638f8  00007ffe`58512fc0 ucrtbase!configure_narrow_argv
00007ff7`b4563900  00007ffe`58518d10 ucrtbase!_seh_filter_exe
00007ff7`b4563908  00007ffe`5856fb70 ucrtbase!register_thread_local_exe_atexit_callback
00007ff7`b4563910  00007ffe`5856fb30 ucrtbase!c_exit
00007ff7`b4563918  00007ffe`58514eb0 ucrtbase!initialize_narrow_environment
00007ff7`b4563920  00007ffe`585174e0 ucrtbase!set_app_type
00007ff7`b4563928  00007ffe`5856fb10 ucrtbase!Exit
00007ff7`b4563930  00007ffe`58512d00 ucrtbase!initialize_onexit_table
00007ff7`b4563938  00007ffe`58509f40 ucrtbase!exit
00007ff7`b4563940  00007ffe`58512d80 ucrtbase!initterm_e
00007ff7`b4563948  00007ffe`58508770 ucrtbase!_errno
00007ff7`b4563950  00007ffe`584ff170 ucrtbase!register_onexit_function
00007ff7`b4563958  00007ffe`58518670 ucrtbase!invalid_parameter_noinfo
00007ff7`b4563960  00007ffe`58512fb0 ucrtbase!get_narrow_winmain_command_line
00007ff7`b4563968  00007ffe`5856c600 ucrtbase!invalid_parameter_noinfo_noreturn
00007ff7`b4563970  00007ffe`5856d470 ucrtbase!terminate
```

```
00007ff7`b4563978  00007ffe`58512d30 ucrtbase!initterm
00007ff7`b4563980  00007ffe`5856fb50 ucrtbase!cexit
00007ff7`b4563988  00000000`00000000
00007ff7`b4563990  00007ffe`585174d0 ucrtbase!_p__commode
00007ff7`b4563998  00007ffe`584fdfc0 ucrtbase!_stdio_common_vsnprintf_s
00007ff7`b45639a0  00007ffe`58516b10 ucrtbase!_set_fmode
00007ff7`b45639a8  00007ffe`58501f10 ucrtbase!__stdio_common_vswprintf
00007ff7`b45639b0  00000000`00000000
00007ff7`b45639b8  00007ffe`5851cab0 ucrtbase!wcsicmp
00007ff7`b45639c0  00007ffe`5851b430 ucrtbase!wcsnlen
00007ff7`b45639c8  00007ffe`584f8fb0 ucrtbase!iswdigit
00007ff7`b45639d0  00000000`00000000
00007ff7`b45639d8  00007ffe`59fbe5d0 shcore!GetDpiForMonitor
00007ff7`b45639e0  00000000`00000000
00007ff7`b45639e8  00007ffe`5a740f00 combase!CoInitializeEx
[onecore\com\combase\class\compobj.cxx @ 3734]
00007ff7`b45639f0  00007ffe`5a7415f0 combase!CoUninitialize
[onecore\com\combase\class\compobj.cxx @ 3793]
00007ff7`b45639f8  00007ffe`5a72ef50 combase!CoCreateGuid
[onecore\com\combase\class\cocrguid.cxx @ 49]
00007ff7`b4563a00  00007ffe`5a753d20 combase!CoCreateFreeThreadedMarshaler
[onecore\com\combase\dcomrem\ipmrshl.cxx @ 201]
00007ff7`b4563a08  00007ffe`5a73d620 combase!CoWaitForMultipleHandles
[onecore\com\combase\dcomrem\sync.cxx @ 86]
00007ff7`b4563a10  00007ffe`5a786640 combase!CoTaskMemAlloc
[onecore\com\combase\class\memapi.cxx @ 437]
00007ff7`b4563a18  00007ffe`5a72c5b0 combase!CoIncrementMTAUsage
[onecore\com\combase\class\compobj.cxx @ 1360]
00007ff7`b4563a20  00007ffe`5a784340 combase!PropVariantClear
[onecore\com\combase\util\propvar.cxx @ 278]
00007ff7`b4563a28  00007ffe`5a7854d0 combase!CoTaskMemFree
[onecore\com\combase\class\memapi.cxx @ 453]
00007ff7`b4563a30  00007ffe`5a743f70 combase!CoCreateInstance
[onecore\com\combase\objact\actapi.cxx @ 252]
00007ff7`b4563a38  00000000`00000000
00007ff7`b4563a40  00007ffe`503c9dc0 urlmon!FindMimeFromData
00007ff7`b4563a48  00000000`00000000
```

7. Now we check Notepad process token (**!token** command) and whether it has impersonating threads:

```
1: kd> !process ffffbe0c870210c0 3f
PROCESS ffffbe0c870210c0
    SessionId: 1  Cid: 1b24    Peb: 4a21ca8000  ParentCid: 1070
    DirBase: 56023002  ObjectTable: ffff800edebca400  HandleCount: 256.
    Image: Notepad.exe
    VadRoot ffffbe0c8b850b60 Vads 109 Clone 0 Private 596. Modified 9. Locked 0.
    DeviceMap ffff800eda518d20
    Token                             ffff800edee53060
    ElapsedTime                       00:02:12.108
    UserTime                          00:00:00.000
    KernelTime                        00:00:00.000
    QuotaPoolUsage[PagedPool]         267152
    QuotaPoolUsage[NonPagedPool]      15472
    Working Set Sizes (now,min,max)  (5965, 50, 345) (23860KB, 200KB, 1380KB)
    PeakWorkingSetSize                5885
    VirtualSize                       4268 Mb
    PeakVirtualSize                   4274 Mb
    PageFaultCount                    6039
    MemoryPriority                    BACKGROUND
    BasePriority                      8
    CommitCharge                      745
    Job                               ffffbe0c8702b580

    PEB at 0000004a21ca8000
    InheritedAddressSpace:    No
    ReadImageFileExecOptions: No
    BeingDebugged:            No
    ImageBaseAddress:         00007ff7b4540000
    NtGlobalFlag:             400
    NtGlobalFlag2:            0
    Ldr                       00007ffe5b0fa120
    Ldr.Initialized:          Yes
    Ldr.InInitializationOrderModuleList: 000001b7f48041c0 . 000001b7f48842e0
    Ldr.InLoadOrderModuleList:           000001b7f4804350 . 000001b7f48842c0
    Ldr.InMemoryOrderModuleList:         000001b7f4804360 . 000001b7f48842d0
                    Base TimeStamp                     Module
            7ff7b4540000 60622ce6 Mar 29 20:39:18 2021 C:\Program
Files\WindowsApps\Microsoft.WindowsNotepad_10.2103.6.0_x64__8wekyb3d8bbwe\Notepad\Notepad.exe
            7ffe5af80000 931cda92 Mar 18 10:55:14 2048 C:\WINDOWS\SYSTEM32\ntdll.dll
            7ffe5a2c0000 7b65e245 Aug 09 13:17:09 2035 C:\WINDOWS\System32\KERNEL32.DLL
            7ffe58a00000 72a6f702 Dec 15 06:00:34 2030 C:\WINDOWS\System32\KERNELBASE.dll
            7ffe5adb0000 5d809272 Sep 17 08:59:46 2019 C:\WINDOWS\System32\SHLWAPI.dll
            7ffe5a160000 90483ed2 Sep 15 20:49:38 2046 C:\WINDOWS\System32\msvcrt.dll
            7ffe59000000 95c2e8f0 Aug 14 19:33:20 2049 C:\WINDOWS\System32\USER32.dll
            7ffe58d80000 2eab7211 Oct 24 09:36:33 1994 C:\WINDOWS\System32\win32u.dll
            7ffe58e20000 0b2998f3 Dec 08 12:58:27 1975 C:\WINDOWS\System32\GDI32.dll
            7ffe588e0000 f03395da Sep 13 13:08:58 2097 C:\WINDOWS\System32\gdi32full.dll
            7ffe58610000 1fb7fd57 Nov 12 03:53:59 1986 C:\WINDOWS\System32\msvcp_win.dll
            7ffe584f0000 00e78ce9 Jun 25 16:14:49 1970 C:\WINDOWS\System32\ucrtbase.dll
            7ffe58e50000 8dfb3d4d Jun 26 02:18:05 2045 C:\WINDOWS\System32\ole32.dll
            7ffe5a6d0000 426c1ced Apr 24 23:25:49 2005 C:\WINDOWS\System32\combase.dll
            7ffe596c0000 7ff0ec4a Jan 07 16:46:02 2038 C:\WINDOWS\System32\RPCRT4.dll
            7ffe5aa90000 b5c44fd4 Aug 20 15:53:08 2066 C:\WINDOWS\System32\COMDLG32.dll
            7ffe59f90000 d40bc30a Sep 25 06:43:38 2082 C:\WINDOWS\System32\shcore.dll
            7ffe597e0000 8cba58e5 Oct 25 16:38:13 2044 C:\WINDOWS\System32\SHELL32.dll
            7ffe5a210000 ce622c7b Sep 21 17:46:51 2079 C:\WINDOWS\System32\ADVAPI32.dll
            7ffe5ad10000 31ec7be5 Jul 17 06:36:37 1996 C:\WINDOWS\System32\sechost.dll
            7ffe5a540000 f6e2d5cf Apr 04 13:30:07 2101 C:\WINDOWS\System32\OLEAUT32.dll
            7ffe44420000 150b8699 Mar 10 12:54:49 1981 C:\WINDOWS\WinSxS\amd64_microsoft.windows.common-
controls_6595b64144ccf1df_6.0.22000.120_none_9d947278b86cc467\COMCTL32.dll
            7ffe561b0000 c2756dbe May 20 04:15:10 2073 C:\WINDOWS\SYSTEM32\PROPSYS.dll
            7ffe50350000 cc1588be Jul 02 05:55:26 2078 C:\WINDOWS\SYSTEM32\urlmon.dll
            7ffe3aa20000 fdebc754 Dec 30 13:40:36 2104 C:\WINDOWS\SYSTEM32\WINSPOOL.DRV
            7ffe46550000 615a9215 Oct 04 06:33:09 2021 C:\Program
Files\WindowsApps\Microsoft.VCLibs.140.00.UWPDesktop_14.0.30704.0_x64__8wekyb3d8bbwe\MSVCP140.dll
            7ffe46540000 615a9218 Oct 04 06:33:12 2021 C:\Program
Files\WindowsApps\Microsoft.VCLibs.140.00.UWPDesktop_14.0.30704.0_x64__8wekyb3d8bbwe\VCRUNTIME140_1.dll
            7ffe46520000 615a9215 Oct 04 06:33:09 2021 C:\Program
Files\WindowsApps\Microsoft.VCLibs.140.00.UWPDesktop_14.0.30704.0_x64__8wekyb3d8bbwe\VCRUNTIME140.dll
            7ffe50090000 5a2fa526 Dec 12 09:45:10 2017 C:\WINDOWS\SYSTEM32\iertutil.dll
            7ffe50050000 35be966e Jul 29 04:26:38 1998 C:\WINDOWS\SYSTEM32\srvcli.dll
            7ffe56f70000 813aa4df Sep 14 20:09:19 2038 C:\WINDOWS\SYSTEM32\netutils.dll
            7ffe5aa50000 356942c7 May 25 11:07:03 1998 C:\WINDOWS\System32\IMM32.DLL
            7ffe58470000 a34302f0 Oct 18 07:57:52 2056 C:\WINDOWS\System32\bcryptPrimitives.dll
            7ffe57570000 fb20135b Jul 06 17:42:03 2103 C:\WINDOWS\SYSTEM32\kernel.appcore.dll
            7ffe55730000 e2c027fe Jul 20 15:26:06 2090 C:\WINDOWS\system32\uxtheme.dll
            7ffe5a620000 1d473905 Jul 26 07:21:57 1985 C:\WINDOWS\System32\clbcatq.dll
            7ffe4ef70000 2eac440d Oct 25 00:32:29 1994 C:\Windows\System32\MrmCoreR.dll
            7ffe4a540000 52a8e73f Dec 11 22:29:19 2013 C:\WINDOWS\SYSTEM32\windows.staterepositoryclient.dll
            7ffe4f670000 41b1e4e8 Dec 04 16:25:12 2004 C:\WINDOWS\SYSTEM32\windows.staterepositorycore.dll
            7ffe583a0000 47c07815 Feb 23 19:46:29 2008 C:\Windows\System32\profapi.dll
            7ffe4ede0000 2a4aa2e7 Jun 26 05:53:59 1992 C:\Windows\System32\Windows.UI.dll
            7ffe4ed50000 f56db9a4 Jun 25 13:14:28 2100 C:\Windows\System32\bcp47mrm.dll
            7ffe51b80000 d6129e9c Oct 23 20:14:36 2083 C:\Windows\System32\twinapi.appcore.dll
            7ffe56470000 b3354271 Apr 10 19:01:21 2065 C:\Windows\System32\WinTypes.dll
            7ffe565e0000 42c927b5 Jul 04 13:12:37 2005 C:\WINDOWS\SYSTEM32\windows.storage.dll
            7ffe5ae10000 81def127 Jan 17 10:06:31 2039 C:\WINDOWS\System32\MSCTF.dll
            7ffe4c1d0000 6627ed04 Apr 23 18:16:52 2024 C:\WINDOWS\SYSTEM32\TextShaping.dll
            7ffe39ac0000 ce6eee78 Oct 01 10:01:44 2079 C:\Windows\System32\efswrt.dll
            7ffe43980000 d4726d59 Dec 12 02:41:29 2082 C:\Windows\System32\oleacc.dll
            7ffe4c6f0000 63938554 Dec 09 18:58:28 2022 C:\WINDOWS\SYSTEM32\textinputframework.dll
            7ffe55280000 9e78ed02 Apr 02 07:45:22 2054 C:\WINDOWS\SYSTEM32\CoreMessaging.dll
            7ffe53340000 6685eb5c Jul 04 01:22:52 2024 C:\WINDOWS\SYSTEM32\CoreUIComponents.dll
            7ffe57c70000 14759998 Nov 16 19:35:52 1980 C:\WINDOWS\SYSTEM32\CRYPTBASE.DLL
    SubSystemData:     00007ffe51dba6e0
    ProcessHeap:       000001b7f4710000
    ProcessParameters: 000001b7f48036e0
    CurrentDirectory:  'C:\Users\dumpa\'
    WindowTitle:  'C:\Program
Files\WindowsApps\Microsoft.WindowsNotepad_10.2103.6.0_x64__8wekyb3d8bbwe\Notepad\Notepad.exe'
```

```
    ImageFile:    'C:\Program
Files\WindowsApps\Microsoft.WindowsNotepad_10.2103.6.0_x64__8wekyb3d8bbwe\Notepad\Notepad.exe'
    CommandLine:  '"C:\Program
Files\WindowsApps\Microsoft.WindowsNotepad_10.2103.6.0_x64__8wekyb3d8bbwe\Notepad\Notepad.exe" '
    DllPath:      'C:\Program Files\WindowsApps\Microsoft.WindowsNotepad_10.2103.6.0_x64__8wekyb3d8bbwe;C:\Program
Files\WindowsApps\Microsoft.VCLibs.140.00.UWPDesktop_14.0.30704.0_x64__8wekyb3d8bbwe;'
    Environment:  000001b7f4802a00
        ALLUSERSPROFILE=C:\ProgramData
        APPDATA=C:\Users\dumpa\AppData\Roaming
        CommonProgramFiles=C:\Program Files\Common Files
        CommonProgramFiles(x86)=C:\Program Files (x86)\Common Files
        CommonProgramW6432=C:\Program Files\Common Files
        COMPUTERNAME=DESKTOP-OGPC0LO
        ComSpec=C:\WINDOWS\system32\cmd.exe
        DriverData=C:\Windows\System32\Drivers\DriverData
        HOMEDRIVE=C:
        HOMEPATH=\Users\dumpa
        LOCALAPPDATA=C:\Users\dumpa\AppData\Local
        LOGONSERVER=\\DESKTOP-OGPC0LO
        NUMBER_OF_PROCESSORS=2
        OneDrive=C:\Users\dumpa\OneDrive
        OS=Windows_NT

Path=C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WINDOWS\System32\WindowsPowerShell\v1.0\;C:\WINDOWS\Sy
stem32\OpenSSH\;C:\Program Files\dotnet\;C:\Program Files
(x86)\dotnet\;C:\Users\dumpa\AppData\Local\Microsoft\WindowsApps;C:\Users\dumpa\.dotnet\tools
        PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
        PROCESSOR_ARCHITECTURE=AMD64
        PROCESSOR_IDENTIFIER=Intel64 Family 6 Model 142 Stepping 10, GenuineIntel
        PROCESSOR_LEVEL=6
        PROCESSOR_REVISION=8e0a
        ProgramData=C:\ProgramData
        ProgramFiles=C:\Program Files
        ProgramFiles(x86)=C:\Program Files (x86)
        ProgramW6432=C:\Program Files
        PSModulePath=C:\Program Files\WindowsPowerShell\Modules;C:\WINDOWS\system32\WindowsPowerShell\v1.0\Modules
        PUBLIC=C:\Users\Public
        SystemDrive=C:
        SystemRoot=C:\WINDOWS
        TEMP=C:\Users\dumpa\AppData\Local\Temp
        TMP=C:\Users\dumpa\AppData\Local\Temp
        USERDOMAIN=DESKTOP-OGPC0LO
        USERDOMAIN_ROAMINGPROFILE=DESKTOP-OGPC0LO
        USERNAME=Training
        USERPROFILE=C:\Users\dumpa
        windir=C:\WINDOWS


        THREAD ffffbe0c89789080  Cid 1b24.10b4  Teb: 0000004a21ca9000 Win32Thread: ffffbe0c8cc9c350 WAIT:
(WrUserRequest) UserMode Non-Alertable
            ffffbe0c8912abc0  QueueObject
        Not impersonating
        DeviceMap                 ffff800eda518d20
        Owning Process            ffffbe0c870210c0       Image:         Notepad.exe
        Attached Process          N/A            Image:         N/A
        Wait Start TickCount      29755          Ticks: 32 (0:00:00:00.500)
        Context Switch Count      1838           IdealProcessor: 1
        UserTime                  00:00:00.031
        KernelTime                00:00:00.109
Unable to load image C:\Program
Files\WindowsApps\Microsoft.WindowsNotepad_10.2103.6.0_x64__8wekyb3d8bbwe\Notepad\Notepad.exe, Win32 error 0n2
*** WARNING: Unable to verify checksum for Notepad.exe
        Win32 Start Address Notepad (0x00007ff7b4560d84)
        Stack Init ffffa28c9fccac70 Current ffffa28c9fcca050
        Base ffffa28c9fccb000 Limit ffffa28c9fcc5000 Call 0000000000000000
        Priority 10 BasePriority 8 PriorityDecrement 0 IoPriority 2 PagePriority 5
        Child-SP          RetAddr               Call Site
        ffffa28c`9fcca090 fffff807`623327f7     nt!KiSwapContext+0x76
        ffffa28c`9fcca1d0 fffff807`623346a9     nt!KiSwapThread+0x3a7
        ffffa28c`9fcca2b0 fffff807`6232e5c4     nt!KiCommitThreadWait+0x159
        ffffa28c`9fcca350 fffff807`6228efe0     nt!KeWaitForSingleObject+0x234
        ffffa28c`9fcca440 ffffbc92`8e76afd6     nt!KeWaitForMultipleObjects+0x540
        ffffa28c`9fcca540 ffffbc92`8e76ac3f     win32kfull!xxxRealSleepThread+0x2c6
        ffffa28c`9fcca660 ffffbc92`8e76e08a     win32kfull!xxxSleepThread2+0xb3
        ffffa28c`9fcca6b0 ffffbc92`8e7b26ec     win32kfull!xxxRealInternalGetMessage+0xc5a
        ffffa28c`9fccaa10 ffffbc92`8dc4645a     win32kfull!NtUserGetMessage+0x8c
        ffffa28c`9fccaaa0 fffff807`62428775     win32k!NtUserGetMessage+0x16
        ffffa28c`9fccaae0 00007ffe`58d81414     nt!KiSystemServiceCopyEnd+0x25 (TrapFrame @ ffffa28c`9fccaae0)
        0000004a`21b9f818 00007ffe`5902464e     win32u!NtUserGetMessage+0x14
        0000004a`21b9f820 00007ff7`b4548208     USER32!GetMessageW+0x2e
        0000004a`21b9f880 00000000`00000000     Notepad+0x8208

        THREAD ffffbe0c8b3e6080  Cid 1b24.2298  Teb: 0000004a21caf000 Win32Thread: 0000000000000000 WAIT: 
(UserRequest) UserMode Non-Alertable
            ffffbe0c89f86560  SynchronizationEvent
            ffffbe0c8a9528e0  SynchronizationEvent
            ffffbe0c8a952b60  SynchronizationEvent
            ffffbe0c89f87be0  SynchronizationEvent
            ffffbe0c89f88b60  SynchronizationEvent
            ffffbe0c8a9525e0  SynchronizationEvent
        Not impersonating
        DeviceMap                 ffff800eda518d20
        Owning Process            ffffbe0c870210c0       Image:         Notepad.exe
        Attached Process          N/A            Image:         N/A
        Wait Start TickCount      21345          Ticks: 8442 (0:00:02:11.906)
        Context Switch Count      2              IdealProcessor: 0
        UserTime                  00:00:00.000
        KernelTime                00:00:00.000
        Win32 Start Address MrmCoreR!Windows::ApplicationModel::Resources::Core::LanguageChangeNotifyThreadProc 
(0x00007ffe4ef9fff0)
        Stack Init ffffa28c9fcfbc70 Current ffffa28c9fcfaee0
        Base ffffa28c9fcfc000 Limit ffffa28c9fcf6000 Call 0000000000000000
        Priority 8 BasePriority 8 PriorityDecrement 0 IoPriority 2 PagePriority 5
        Child-SP          RetAddr               Call Site
        ffffa28c`9fcfaf20 fffff807`623327f7     nt!KiSwapContext+0x76
        ffffa28c`9fcfb060 fffff807`623346a9     nt!KiSwapThread+0x3a7
        ffffa28c`9fcfb140 fffff807`6228ed51     nt!KiCommitThreadWait+0x159
        ffffa28c`9fcfb1e0 fffff807`627702c5     nt!KeWaitForMultipleObjects+0x2b1
        ffffa28c`9fcfb2e0 fffff807`62672b79     nt!ObWaitForMultipleObjects+0x2d5
        ffffa28c`9fcfb7e0 fffff807`62428775     nt!NtWaitForMultipleObjects+0x119
        ffffa28c`9fcfba70 00007ffe`5b0242a4     nt!KiSystemServiceCopyEnd+0x25 (TrapFrame @ ffffa28c`9fcfbae0)
        0000004a`220ff668 00007ffe`58a4fb10     ntdll!NtWaitForMultipleObjects+0x14
        0000004a`220ff670 00007ffe`5a79e185     KERNELBASE!WaitForMultipleObjectsEx+0xf0
        0000004a`220ff960 00007ffe`5a73d6a0     combase!DefaultWaitForHandles+0x45 
[onecore\com\combase\dcomrem\sync.cxx @ 38]
        0000004a`220ff9c0 00007ffe`4efa0681     combase!CoWaitForMultipleHandles+0x80 
[onecore\com\combase\dcomrem\sync.cxx @ 123]
        0000004a`220ffa00 00007ffe`5a2d54e0     
MrmCoreR!Windows::ApplicationModel::Resources::Core::LanguageChangeNotifyThreadProc+0x691
        0000004a`220ffbf0 00007ffe`5af8485b     KERNEL32!BaseThreadInitThunk+0x10
        0000004a`220ffc20 00000000`00000000     ntdll!RtlUserThreadStart+0x2b

        THREAD ffffbe0c8b3e5080  Cid 1b24.229c  Teb: 0000004a21cb1000 Win32Thread: 0000000000000000 WAIT: (WrQueue) 
UserMode Alertable
            ffffbe0c8b641240  QueueObject
        Not impersonating
        DeviceMap                 ffff800eda518d20
        Owning Process            ffffbe0c870210c0       Image:         Notepad.exe
        Attached Process          N/A            Image:         N/A
        Wait Start TickCount      21350          Ticks: 8437 (0:00:02:11.828)
        Context Switch Count      3              IdealProcessor: 1
        UserTime                  00:00:00.000
        KernelTime                00:00:00.000
        Win32 Start Address ntdll!TppWorkerThread (0x00007ffe5af96950)
        Stack Init ffffa28c9fd09c70 Current ffffa28c9fd09360
        Base ffffa28c9fd0a000 Limit ffffa28c9fd04000 Call 0000000000000000
        Priority 9 BasePriority 8 PriorityDecrement 16 IoPriority 2 PagePriority 5
        Child-SP          RetAddr               Call Site
        ffffa28c`9fd093a0 fffff807`623327f7     nt!KiSwapContext+0x76
        ffffa28c`9fd094e0 fffff807`623346a9     nt!KiSwapThread+0x3a7
        ffffa28c`9fd095c0 fffff807`62337106     nt!KiCommitThreadWait+0x159
        ffffa28c`9fd09660 fffff807`62336b18     nt!KeRemoveQueueEx+0x2b6
        ffffa28c`9fd09710 fffff807`6233937c     nt!IoRemoveIoCompletion+0x98
        ffffa28c`9fd09830 fffff807`62428775     nt!NtWaitForWorkViaWorkerFactory+0x39c
        ffffa28c`9fd09a70 00007ffe`5b027304     nt!KiSystemServiceCopyEnd+0x25 (TrapFrame @ ffffa28c`9fd09ae0)
        0000004a`221ff8f8 00007ffe`5af96c2f     ntdll!NtWaitForWorkViaWorkerFactory+0x14
        0000004a`221ff900 00007ffe`5a2d54e0     ntdll!TppWorkerThread+0x2df
        0000004a`221ffbf0 00007ffe`5af8485b     KERNEL32!BaseThreadInitThunk+0x10
        0000004a`221ffc20 00000000`00000000     ntdll!RtlUserThreadStart+0x2b

        THREAD ffffbe0c8b3e3080  Cid 1b24.22a4  Teb: 0000004a21cb5000 Win32Thread: 0000000000000000 WAIT: (WrQueue) 
UserMode Alertable
```

```
            ffffbe0c8b641240  QueueObject
        Not impersonating
        DeviceMap                 ffff800eda518d20
        Owning Process            ffffbe0c870210c0       Image:         Notepad.exe
        Attached Process          N/A            Image:         N/A
        Wait Start TickCount      25196          Ticks: 4591 (0:00:01:11.734)
        Context Switch Count      4              IdealProcessor: 1
        UserTime                  00:00:00.000
        KernelTime                00:00:00.000
        Win32 Start Address ntdll!TppWorkerThread (0x00007ffe5af96950)
        Stack Init ffffa28c9fd6bc70 Current ffffa28c9fd6b360
        Base ffffa28c9fd6c000 Limit ffffa28c9fd66000 Call 0000000000000000
        Priority 8 BasePriority 8 PriorityDecrement 0 IoPriority 2 PagePriority 5
        Child-SP          RetAddr               Call Site
        ffffa28c`9fd6b3a0 fffff807`623327f7     nt!KiSwapContext+0x76
        ffffa28c`9fd6b4e0 fffff807`623346a9     nt!KiSwapThread+0x3a7
        ffffa28c`9fd6b5c0 fffff807`62337106     nt!KiCommitThreadWait+0x159
        ffffa28c`9fd6b660 fffff807`62336b18     nt!KeRemoveQueueEx+0x2b6
        ffffa28c`9fd6b710 fffff807`6233937c     nt!IoRemoveIoCompletion+0x98
        ffffa28c`9fd6b830 fffff807`62428775     nt!NtWaitForWorkViaWorkerFactory+0x39c
        ffffa28c`9fd6ba70 00007ffe`5b027304     nt!KiSystemServiceCopyEnd+0x25 (TrapFrame @ ffffa28c`9fd6bae0)
        0000004a`223ffbc8 00007ffe`5af96c2f     ntdll!NtWaitForWorkViaWorkerFactory+0x14
        0000004a`223ffbd0 00007ffe`5a2d54e0     ntdll!TppWorkerThread+0x2df
        0000004a`223ffec0 00007ffe`5af8485b     KERNEL32!BaseThreadInitThunk+0x10
        0000004a`223ffef0 00000000`00000000     ntdll!RtlUserThreadStart+0x2b
```

```
1: kd> !token ffff800edee53060
_TOKEN 0xffff800edee53060
TS Session ID: 0x1
User: S-1-5-21-3407489871-1359576761-456439074-1001
User Groups:
 00 S-1-5-21-3407489871-1359576761-456439074-513
    Attributes - Mandatory Default Enabled
 01 S-1-1-0
    Attributes - Mandatory Default Enabled
 02 S-1-5-114
    Attributes - DenyOnly
 03 S-1-5-32-544
    Attributes - DenyOnly
 04 S-1-5-32-545
    Attributes - Mandatory Default Enabled
 05 S-1-5-4
    Attributes - Mandatory Default Enabled
 06 S-1-2-1
    Attributes - Mandatory Default Enabled
 07 S-1-5-11
    Attributes - Mandatory Default Enabled
 08 S-1-5-15
    Attributes - Mandatory Default Enabled
 09 S-1-5-113
    Attributes - Mandatory Default Enabled
 10 S-1-5-5-0-434421
    Attributes - Mandatory Default Enabled LogonId
 11 S-1-2-0
    Attributes - Mandatory Default Enabled
 12 S-1-5-64-10
    Attributes - Mandatory Default Enabled
 13 S-1-16-8192
    Attributes - GroupIntegrity GroupIntegrityEnabled
Primary Group: S-1-5-21-3407489871-1359576761-456439074-513
Privs:
 19 0x000000013 SeShutdownPrivilege               Attributes -
 23 0x000000017 SeChangeNotifyPrivilege           Attributes - Enabled Default
 25 0x000000019 SeUndockPrivilege                 Attributes -
 33 0x000000021 SeIncreaseWorkingSetPrivilege     Attributes -
 34 0x000000022 SeTimeZonePrivilege               Attributes -
Authentication ID:         (0,6a25a)
Impersonation Level:       Anonymous
TokenType:                 Primary
Source: User32             TokenFlags: 0x2a00 ( Token in use )
Token ID: 1a5b91           ParentToken ID: 6a25d
Modified ID:               (0, 1a5ae9)
RestrictedSidCount: 0      RestrictedSids: 0x0000000000000000
OriginatingLogonSession: 3e7
PackageSid: (null)
CapabilityCount: 0      Capabilities: 0x0000000000000000
LowboxNumberEntry: 0x0000000000000000
Security Attributes:
Unable to get the offset of nt!_AUTHZBASEP_SECURITY_ATTRIBUTE.ListLink
Process Token TrustLevelSid: (null)
```

8. To check for hidden processes and drivers, we can dump all kernel pool entries having *Proc* and *Driv* tags (**!poolfind** command) and then find out any discrepancy with the active process list (**!process 0 0**), for example.

```
1: kd> !poolfind Proc

Scanning large pool allocation table for tag 0x636f7250 (Proc) (ffffbe0c86240000 :
ffffbe0c86340000)

ffffbe0c8ac07010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8a7a6010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8a0c4010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8bfb1060 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c876f8010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8a55b060 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89f5e010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89aae010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c840eb000 : tag Proc, size    0x1000, Nonpaged pool
ffffbe0c889b1150 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89a06010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c888c9090 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8b224010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8b4b2010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8be76050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8ad81050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c898c1050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8b417010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c888cb070 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c876f2010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8a455010 : tag Proc, size     0xe70, Nonpaged pool

Searching nonpaged pool (ffffbe0000000000 : ffffce0000000000) for tag 0x636f7250 (Proc)

ffffbe0c84120010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c84135010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c84136020 : tag Proc, size     0xdf0, Nonpaged pool
ffffbe0c84185010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c841b0010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c841ba010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c841ed010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c84c99050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c84caf010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c87021050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c87648050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c876f2010 : tag Proc, size     0xe70, Nonpaged pool
```

```
ffffbe0c876f8010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c877ec010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8780f010 : tag Proc, size     0xdf0, Nonpaged pool
ffffbe0c87f2b010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c887d4010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c88884050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c888c9090 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c888cb070 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c888dc010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c889ae010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c889af150 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c889b1150 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c889c7010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89013050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89042010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89102050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89120010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8919e010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c891a0010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c891a8050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c891b9050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c891c6010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89209010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89241010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89255050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c892b6010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c892bc010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89348010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8934d010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89362010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89363010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c893bc010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c893bf010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89407010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8940a010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89496010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89534050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89538010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89596010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c895b3010 : tag Proc, size     0xdf0, Nonpaged pool
ffffbe0c895cc010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c895df010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8963c010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c896b1010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c896d2050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89728010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89730010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89745050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c897da010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c897dc050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c897ed050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89895010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c898c1050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c899e8050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89a02050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89a06010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89a07010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89a0a010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89a42010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89a8e050 : tag Proc, size     0xe70, Nonpaged pool
```

```
ffffbe0c89a96010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89a9b010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89aa6010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89aa9010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89aaa010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89aae010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89ac1050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89c30010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89cf8010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89e84050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c89f5e010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8a054050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8a071010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8a0c4010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8a0cb010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8a232050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8a334050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8a337010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8a402050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8a43b010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8a454010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8a455010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8a4e9010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8a4ef050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8a4f7010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8a4f8010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8a6c1010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8a729010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8a7a6010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8a854050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8a859010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8a924050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8a982010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8aa02050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8aa37010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8aaa0050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8ab52010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8ac07010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8ac8e010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8ac98010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8acd6010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8ad84010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8ad94010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8ae6f010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8ae72010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8ae76050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8aedd010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8aee1010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8aeeb050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8af24050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8af46050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8af55010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8b092010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8b0d6050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8b208050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8b224010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8b317010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8b318010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8b37b010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8b396050 : tag Proc, size     0xe70, Nonpaged pool
```

```
ffffbe0c8b3d5010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8b402050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8b417010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8b49a010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8b4b2010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8b4b6050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8b4d6010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8b4d8050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8b4e9050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8b538010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8b5b3050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8b5e6050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8b696050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8b887050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8be62050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8bed8010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8c2cd050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8c2de050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8c576050 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8c9d3010 : tag Proc, size     0xe70, Nonpaged pool
ffffbe0c8cce9050 : tag Proc, size     0xe70, Nonpaged pool
```

Let's check the last *Proc* entry:

```
1: kd> dc ffffbe0c8cce9050-10 L50
ffffbe0c`8cce9040  02e80000 636f7250 00000000 00000000  ....Proc........
ffffbe0c`8cce9050  00000000 00000000 00000000 00000000  ................
ffffbe0c`8cce9060  00000000 00000000 00000000 00000020  ............ ...
ffffbe0c`8cce9070  00001000 00000d88 00000078 00000000  ........x.......
ffffbe0c`8cce9080  8900b580 ffffbe0c 00000000 00000000  ................
ffffbe0c`8cce9090  00057e01 00000000 0000000e 00000000  .~..............
ffffbe0c`8cce90a0  00000000 00000000 00880003 00000000  ................
ffffbe0c`8cce90b0  8900b580 ffffbe0c dac9a1af ffff800e  ................
ffffbe0c`8cce90c0  00000003 00000000 8cd0ebd0 ffffbe0c  ................
ffffbe0c`8cce90d0  8cd18080 ffffbe0c 8cce90d8 ffffbe0c  ................
ffffbe0c`8cce90e0  8cce90d8 ffffbe0c 869f5002 00000000  .........P......
ffffbe0c`8cce90f0  8b2f4378 ffffbe0c 8760a378 ffffbe0c  xC/.....x.`.....
ffffbe0c`8cce9100  00000000 00000000 00000000 00000000  ................
ffffbe0c`8cce9110  00200001 00000000 00000003 00000000  .. .............
ffffbe0c`8cce9120  00000000 00000000 00000000 00000000  ................
ffffbe0c`8cce9130  00000000 00000000 00000000 00000000  ................
ffffbe0c`8cce9140  00000000 00000000 00000000 00000000  ................
ffffbe0c`8cce9150  00000000 00000000 00000000 00000000  ................
ffffbe0c`8cce9160  00000000 00000000 00000000 00000000  ................
ffffbe0c`8cce9170  00000000 00000000 00000000 00000000  ................

1: kd> !process ffffbe0c`8cce90c0 0
PROCESS ffffbe0c8cce90c0
    SessionId: 1  Cid: 1200    Peb: 7a067c000  ParentCid: 0f0c
    DirBase: 869f5002  ObjectTable: ffff800edd1bbc80  HandleCount: 335.
    Image: msedge.exe
```

Let's check some *Driv* entry:

```
1: kd> !poolfind Driv

Scanning large pool allocation table for tag 0x76697244 (Driv) (ffffbe0c86240000 : 
ffffbe0c86340000)

ffffbe0c84f48d70 : tag Driv, size     0x1f0, Nonpaged pool
```

```
ffffbe0c89041de0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c87e0adc0 : tag Driv, size     0x210, Nonpaged pool
ffffbe0c84a9cac0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c87d8a7a0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c87cc4de0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84b3b9d0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84dce030 : tag Driv, size     0x970, Nonpaged pool
ffffbe0c84dd6d30 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c89a6a330 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c8787ed50 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84bd9a20 : tag Driv, size     0x5b0, Nonpaged pool
ffffbe0c84eb8100 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84f3d030 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84f3d7e0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84f3ddb0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c8782b0f0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c8b5cede0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84dd2000 : tag Driv, size    0x2710, Nonpaged pool

Searching nonpaged pool (ffffbe0000000000 : ffffce0000000000) for tag 0x76697244 (Driv)

ffffbe0c84076de0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84113de0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84115de0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84119de0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c8411dde0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c8412dde0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c8412fde0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84131de0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84138de0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c8413cde0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c8413ed30 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84142d30 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84147d30 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84168e00 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c8416ae00 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84175de0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c8417cab0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c8417de00 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84180de0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84186ab0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84187e00 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84189d80 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c8418bab0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c842b66f0 : tag Driv, size      0x20, Nonpaged pool
ffffbe0c842b6840 : tag Driv, size      0x20, Nonpaged pool
ffffbe0c842b6ae0 : tag Driv, size      0x20, Nonpaged pool
ffffbe0c846f96c0 : tag Driv, size      0x20, Nonpaged pool
ffffbe0c846f98d0 : tag Driv, size      0x20, Nonpaged pool
ffffbe0c846f9900 : tag Driv, size      0x20, Nonpaged pool
ffffbe0c846f9930 : tag Driv, size      0x20, Nonpaged pool
ffffbe0c8486ac40 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c8486ec40 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84949010 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c8494dd70 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84961d80 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84971db0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84973d70 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84977b50 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84983aa0 : tag Driv, size     0x1f0, Nonpaged pool
```

```
ffffbe0c84983cb0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c8498ecf0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c849b7b70 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c849b7d80 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c849c3d40 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c849c4de0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c849c9c40 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c849ca5e0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c849e1c80 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c849e28a0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c849e2ab0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c849e2cc0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84a918e0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84a92b70 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84a93740 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84a93d30 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84a94da0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84a96010 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84a98010 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84a98220 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84a9e450 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84a9ea10 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84abc930 : tag Driv, size      0x60, Nonpaged pool
ffffbe0c84acc470 : tag Driv, size      0x40, Nonpaged pool
ffffbe0c84bcab80 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84bcb9a0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84bd3aa0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84bd4d00 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84bd6b30 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84c79ad0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84cd8010 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84cd8220 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84cdfb20 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84d82de0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84da9a60 : tag Driv, size      0x20, Nonpaged pool
ffffbe0c84db89e0 : tag Driv, size     0x3a0, Nonpaged pool
ffffbe0c84dd6d30 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84dd7df0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84decc80 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84df1010 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84df1240 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84df74e0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84df8de0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84eafde0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84eb9010 : tag Driv, size     0x5b0, Nonpaged pool
ffffbe0c84f35380 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84f35d20 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84f36290 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84f36c50 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84f37700 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84f3b2b0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84f3ec40 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84f3f280 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84f3f700 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84f40dd0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84f4daa0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c84f51010 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c8757f050 : tag Driv, size     0x140, Nonpaged pool
ffffbe0c878052c0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c87811320 : tag Driv, size     0x1f0, Nonpaged pool
```

```
ffffbe0c878115b0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c87823d90 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c87826da0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c8782c9b0 : tag Driv, size     0x530, Nonpaged pool
ffffbe0c8782e010 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c878528f0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c87852dc0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c87870dd0 : tag Driv, size     0x1f0, Nonpaged pool
ffffbe0c878b4ce0 : tag Driv, size     0x1f0, Nonpaged pool
[...]

1: kd> dc ffffbe0c849b7b70-10 L50
ffffbe0c`849b7b60  02200000 76697244 00000000 00000000  .. .Driv........
ffffbe0c`849b7b70  d49395e0 ffff800e 000c000c 00000000  ................
ffffbe0c`849b7b80  d4dff0f0 ffff800e 00000000 00000000  ................
ffffbe0c`849b7b90  00000012 00000000 00000000 00000000  ................
ffffbe0c`849b7ba0  00000000 00000000 120200cb 00000000  ................
ffffbe0c`849b7bb0  00000001 00000000 d49facef ffff800e  ................
ffffbe0c`849b7bc0  01500004 00000000 87ed1d40 ffffbe0c  ..P.....@.......
ffffbe0c`849b7bd0  00000012 00000000 66320000 fffff807  ..........2f....
ffffbe0c`849b7be0  00073000 00000000 8406de00 ffffbe0c  .0..............
ffffbe0c`849b7bf0  849b7d10 ffffbe0c 00240024 00000000  .}......$.$.....
ffffbe0c`849b7c00  849fc9d0 ffffbe0c 62d3d700 fffff807  ...........b....
ffffbe0c`849b7c10  84a489c0 ffffbe0c 66389010 fffff807  ..........8f....
ffffbe0c`849b7c20  00000000 00000000 00000000 00000000  ................
ffffbe0c`849b7c30  6635d660 fffff807 6635d660 fffff807  `.5f....`.5f....
ffffbe0c`849b7c40  66325870 fffff807 66325870 fffff807  pX2f....pX2f....
ffffbe0c`849b7c50  66325870 fffff807 66325870 fffff807  pX2f....pX2f....
ffffbe0c`849b7c60  66325870 fffff807 66325870 fffff807  pX2f....pX2f....
ffffbe0c`849b7c70  66325870 fffff807 66325870 fffff807  pX2f....pX2f....
ffffbe0c`849b7c80  66325870 fffff807 66325870 fffff807  pX2f....pX2f....
ffffbe0c`849b7c90  66325870 fffff807 6635e0c0 fffff807  pX2f......5f..

1: kd> !drvobj ffffbe0c`849b7bc0
Driver object (ffffbe0c849b7bc0) is for:
 \FileSystem\FltMgr

Driver Extension List: (id , addr)

Device Object list:
ffffbe0c87ed1d40  ffffbe0c87d07ac0  ffffbe0c84f3f040  ffffbe0c84f3b040
ffffbe0c84f3e040  ffffbe0c84eba040  ffffbe0c84cd8510  ffffbe0c84eaec00
ffffbe0c84dc08d0  ffffbe0c84dc0b40  ffffbe0c849d2d80  ffffbe0c849c32e0
ffffbe0c849c3040  ffffbe0c849b7500  ffffbe0c849b72d0
```

Note that another approach is to dump all handles of Process type from System process:

```
1: kd> !process 0 0 System
PROCESS ffffbe0c840eb040
    SessionId: none  Cid: 0004    Peb: 00000000  ParentCid: 0000
    DirBase: 001ae002  ObjectTable: ffff800ed4820c80  HandleCount: 3961.
    Image: System

1: kd> !handle 0 3 ffffbe0c840eb040 Process

Searching for handles of type Process

PROCESS ffffbe0c840eb040
    SessionId: none  Cid: 0004    Peb: 00000000  ParentCid: 0000
    DirBase: 001ae002  ObjectTable: ffff800ed4820c80  HandleCount: 3961.
    Image: System

Kernel handle table at ffff800ed4820c80 with 3961 entries in use

0004: Object: ffffbe0c840eb040  GrantedAccess: 001fffff (Protected) Entry: ffff800ed48ac010
Object: ffffbe0c840eb040  Type: (ffffbe0c840c6900) Process
    ObjectHeader: ffffbe0c840eb010 (new version)
        HandleCount: 4  PointerCount: 131126

0050: Object: ffffbe0c84136080  GrantedAccess: 001fffff (Protected) (Audit) Entry:
ffff800ed48ac140
Object: ffffbe0c84136080  Type: (ffffbe0c840c6900) Process
    ObjectHeader: ffffbe0c84136050 (new version)
        HandleCount: 1  PointerCount: 44875

[...]

4020: Object: ffffbe0c8b318080  GrantedAccess: 001fffff (Protected) (Audit) Entry:
ffff800edc025080
Object: ffffbe0c8b318080  Type: (ffffbe0c840c6900) Process
    ObjectHeader: ffffbe0c8b318050 (new version)
        HandleCount: 9  PointerCount: 294575

4058: Object: ffffbe0c8a7a6080  GrantedAccess: 0000102a (Protected) (Audit) Entry:
ffff800edc025160
Object: ffffbe0c8a7a6080  Type: (ffffbe0c840c6900) Process
    ObjectHeader: ffffbe0c8a7a6050 (new version)
        HandleCount: 11  PointerCount: 359306

4060: Object: ffffbe0c8c9d3080  GrantedAccess: 001fffff (Protected) (Audit) Entry:
ffff800edc025180
Object: ffffbe0c8c9d3080  Type: (ffffbe0c840c6900) Process
    ObjectHeader: ffffbe0c8c9d3050 (new version)
        HandleCount: 5  PointerCount: 163481

1: kd> !process ffffbe0c8c9d3080 0
PROCESS ffffbe0c8c9d3080
    SessionId: 1  Cid: 2560    Peb: de04a58000  ParentCid: 1070
    DirBase: 3bce9002  ObjectTable: ffff800edffa9f00  HandleCount:  70.
    Image: cmd.exe
```

9. And finally, we check I/O stack traces for all IRPs (a verbose form of the **!irpfind** command):

```
0: kd> !irpfind -v

[...]

ffffbe0c897a6aa0: Irp is active with 12 stacks 11 is current (= 0xffffbe0c897a6e40)
 No Mdl: No System Buffer: Thread ffffbe0c8aef9080:  Irp stack trace.
     cmd  flg cl Device   File     Completion-Context
 [N/A(0), N/A(0)]
            0  0 00000000 00000000 00000000-00000000

                        Args: 00000000 00000000 00000000 00000000
 [N/A(0), N/A(0)]
            0  0 00000000 00000000 00000000-00000000

                        Args: 00000000 00000000 00000000 00000000
 [N/A(0), N/A(0)]
            0  0 00000000 00000000 00000000-00000000
```

```
                      Args: 00000000 00000000 00000000 00000000
 [N/A(0), N/A(0)]
            0  0 00000000 00000000 00000000-00000000

                      Args: 00000000 00000000 00000000 00000000
 [N/A(0), N/A(0)]
            0  0 00000000 00000000 00000000-00000000

                      Args: 00000000 00000000 00000000 00000000
 [N/A(0), N/A(0)]
            0  0 00000000 00000000 00000000-00000000

                      Args: 00000000 00000000 00000000 00000000
 [N/A(0), N/A(0)]
            0  0 00000000 00000000 00000000-00000000

                      Args: 00000000 00000000 00000000 00000000
 [N/A(0), N/A(0)]
            0  0 00000000 00000000 00000000-00000000

                      Args: 00000000 00000000 00000000 00000000
 [N/A(0), N/A(0)]
            0  0 00000000 00000000 00000000-00000000

                      Args: 00000000 00000000 00000000 00000000
 [N/A(0), N/A(0)]
            0  0 00000000 00000000 00000000-00000000

                      Args: 00000000 00000000 00000000 00000000
>[IRP_MJ_DIRECTORY_CONTROL(c), N/A(2)]
            1 e1 ffffbe0c84c92030 ffffbe0c8aaecc70 fffff80766325400-ffffbe0c8b1c2520 Success Error Cancel
pending
              \FileSystem\Ntfs      FLTMGR!FltpPassThroughCompletion
                      Args: 00000020 0000011f 00000000 00000000
 [IRP_MJ_DIRECTORY_CONTROL(c), N/A(2)]
            1  0 ffffbe0c84dc08d0 ffffbe0c8aaecc70 00000000-00000000
              \FileSystem\FltMgr
                      Args: 00000020 0000011f 00000000 00000000

[...]

ffffbe0c84f508a0: Irp is active with 12 stacks 8 is current (= 0xffffbe0c84f50b68)
 No Mdl: No System Buffer: Thread 00000000:  Irp stack trace.
     cmd  flg cl Device   File     Completion-Context
 [N/A(0), N/A(0)]
            0  2 00000000 00000000 00000000-00000000

                      Args: 00000000 00000000 00000000 ffffffffc0000120
 [N/A(0), N/A(0)]
            0  0 00000000 00000000 00000000-00000000

                      Args: 00000000 00000000 00000000 00000000
 [N/A(0), N/A(0)]
            0  0 00000000 00000000 00000000-00000000

                      Args: 00000000 00000000 00000000 00000000
 [N/A(0), N/A(0)]
            0  0 00000000 00000000 00000000-00000000

                      Args: 00000000 00000000 00000000 00000000
 [N/A(0), N/A(0)]
            0  0 00000000 00000000 00000000-00000000
```

```
                      Args: 00000000 00000000 00000000 00000000
 [N/A(0), N/A(0)]
            0  0 00000000 00000000 00000000-00000000

                      Args: 00000000 00000000 00000000 00000000
 [N/A(0), N/A(0)]
            0  0 00000000 00000000 00000000-00000000

                      Args: 00000000 00000000 00000000 00000000
>[IRP_MJ_INTERNAL_DEVICE_CONTROL(f), N/A(0)]
            0  1 ffffbe0c8784e060 00000000 00000000-00000000    pending
              \Driver\USBXHCI
                      Args: ffffbe0c87cea600 00000000 0x220003 00000000
 [IRP_MJ_INTERNAL_DEVICE_CONTROL(f), N/A(0)]
            0 e0 ffffbe0c87826060 00000000 00000000-00000000
              \Driver\USBXHCI
                      Args: ffffbe0c87cea600 00000000 0x220003 00000000
 [IRP_MJ_INTERNAL_DEVICE_CONTROL(f), N/A(0)]
            0 e1 ffffbe0c87826060 00000000 00000000-00000000    pending
              \Driver\USBXHCI
                      Args: ffffbe0c87cea600 00000000 0x220003 00000000
 [IRP_MJ_INTERNAL_DEVICE_CONTROL(f), N/A(0)]
            0 e0 ffffbe0c87cc4670 00000000 fffff80769582be0-ffffbe0c87cea600 Success Error Cancel
              \Driver\USBHUB3       hidusb!HumReadCompletion
                      Args: ffffbe0c87cea600 00000000 0x220003 00000000
 [IRP_MJ_INTERNAL_DEVICE_CONTROL(f), N/A(0)]
            0 e0 ffffbe0c87cf74b0 00000000 fffff807695aa620-ffffbe0c87cf7620 Success Error Cancel
              \Driver\HidUsb HIDCLASS!HidpInterruptReadComplete
                      Args: 00000032 00000000 0xb000b 00000000

[...]
```

If any entry is suspicious, you can check its Device and File fields using the **!devobj** and **!fileobj** commands.

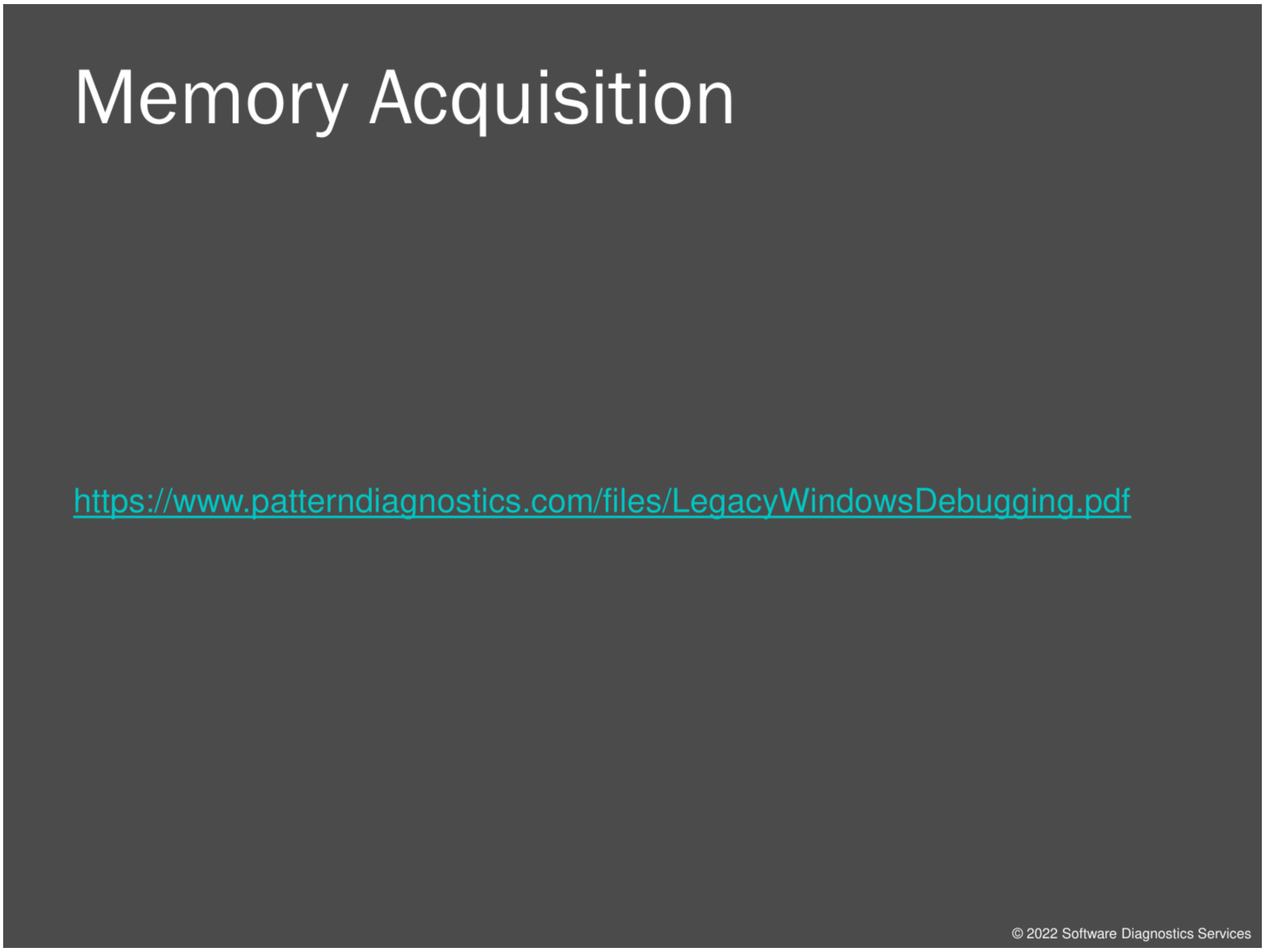
10. If you know the Device object address, you can check handles that reference it:

```
1: kd> !devhandles ffffbe0c84c92030

Checking handle table for process 0xffffbe0c840eb040
Kernel handle table at ffff800ed4820c80 with 3961 entries in use

PROCESS ffffbe0c840eb040
    SessionId: none  Cid: 0004    Peb: 00000000  ParentCid: 0000
    DirBase: 001ae002  ObjectTable: ffff800ed4820c80  HandleCount: 3961.
    Image: System

0970: Object: ffffbe0c8897a210  GrantedAccess: 001f0006 (Inherit) (Audit) Entry: ffff800ed87fd5c0
Object: ffffbe0c8897a210  Type: (ffffbe0c840fe7a0) File
    ObjectHeader: ffffbe0c8897a1e0 (new version)
        HandleCount: 1  PointerCount: 32768
        Directory Object: 00000000  Name: \Sessions\1\AppContainerNamedObjects\S-1-15-2-95739096-
486727260-2033287795-3853587803-1685597119-444378811-2746676523 {NamedPipe}

PROCESS ffffbe0c840eb040
    SessionId: none  Cid: 0004    Peb: 00000000  ParentCid: 0000
    DirBase: 001ae002  ObjectTable: ffff800ed4820c80  HandleCount: 3961.
    Image: System
```

```
0978: Object: ffffbe0c8897a6c0  GrantedAccess: 001f0006 (Protected) Entry: ffff800ed87fd5e0
Object: ffffbe0c8897a6c0  Type: (ffffbe0c840fe7a0) File
    ObjectHeader: ffffbe0c8897a690 (new version)
        HandleCount: 1  PointerCount: 32768
        Directory Object: 00000000  Name: \Sessions\0\AppContainerNamedObjects\S-1-15-2-95739096-
486727260-2033287795-3853587803-1685597119-444378811-2746676523 {NamedPipe}

PROCESS ffffbe0c840eb040
    SessionId: none  Cid: 0004    Peb: 00000000  ParentCid: 0000
    DirBase: 001ae002  ObjectTable: ffff800ed4820c80  HandleCount: 3961.
    Image: System

175c: Object: ffffbe0c8999db00  GrantedAccess: 0012019f (Protected) (Audit) Entry: ffff800ed95f6d70
Object: ffffbe0c8999db00  Type: (ffffbe0c840fe7a0) File
    ObjectHeader: ffffbe0c8999dad0 (new version)
        HandleCount: 1  PointerCount: 32769
        Directory Object: 00000000  Name: \ {NamedPipe}

PROCESS ffffbe0c840eb040
    SessionId: none  Cid: 0004    Peb: 00000000  ParentCid: 0000
    DirBase: 001ae002  ObjectTable: ffff800ed4820c80  HandleCount: 3961.
    Image: System

1e0c: Object: ffffbe0c89e5d390  GrantedAccess: 001f0006 (Inherit) (Audit) Entry: ffff800ed7135830
Object: ffffbe0c89e5d390  Type: (ffffbe0c840fe7a0) File
    ObjectHeader: ffffbe0c89e5d360 (new version)
        HandleCount: 1  PointerCount: 32768
        Directory Object: 00000000  Name: \Sessions\0\AppContainerNamedObjects\S-1-15-2-4197891166-
2373215845-1024567249-2215767161-3850818010-3023594601-3129579408 {NamedPipe}

PROCESS ffffbe0c840eb040
    SessionId: none  Cid: 0004    Peb: 00000000  ParentCid: 0000
    DirBase: 001ae002  ObjectTable: ffff800ed4820c80  HandleCount: 3961.
    Image: System

240c: Object: ffffbe0c8a8793c0  GrantedAccess: 001f0006 (Protected) Entry: ffff800edb92f030
Object: ffffbe0c8a8793c0  Type: (ffffbe0c840fe7a0) File
    ObjectHeader: ffffbe0c8a879390 (new version)
        HandleCount: 1  PointerCount: 32768
        Directory Object: 00000000  Name: \Sessions\1\AppContainerNamedObjects\S-1-15-2-283421221-
3183566570-1718213290-751554359-3541592344-2312209569-3374928651 {NamedPipe}

PROCESS ffffbe0c840eb040
    SessionId: none  Cid: 0004    Peb: 00000000  ParentCid: 0000
    DirBase: 001ae002  ObjectTable: ffff800ed4820c80  HandleCount: 3961.
    Image: System

2450: Object: ffffbe0c8a87a9a0  GrantedAccess: 001f0006 (Protected) (Inherit) (Audit) Entry:
ffff800edb92f140
Object: ffffbe0c8a87a9a0  Type: (ffffbe0c840fe7a0) File
    ObjectHeader: ffffbe0c8a87a970 (new version)
        HandleCount: 1  PointerCount: 32768
        Directory Object: 00000000  Name: \Sessions\1\AppContainerNamedObjects\S-1-15-2-515815643-
2845804217-1874292103-218650560-777617685-4287762684-137415000 {NamedPipe}

PROCESS ffffbe0c840eb040
    SessionId: none  Cid: 0004    Peb: 00000000  ParentCid: 0000
    DirBase: 001ae002  ObjectTable: ffff800ed4820c80  HandleCount: 3961.
    Image: System

24bc: Object: ffffbe0c8a88d0f0  GrantedAccess: 001f0006 (Audit) Entry: ffff800edb92f2f0
Object: ffffbe0c8a88d0f0  Type: (ffffbe0c840fe7a0) File
    ObjectHeader: ffffbe0c8a88d0c0 (new version)
        HandleCount: 1  PointerCount: 32768
        Directory Object: 00000000  Name: \Sessions\1\AppContainerNamedObjects\S-1-15-2-1726375552-
1729233799-74693324-3851689839-2151781990-3623637752-3611872497 {NamedPipe}

PROCESS ffffbe0c840eb040
    SessionId: none  Cid: 0004    Peb: 00000000  ParentCid: 0000
    DirBase: 001ae002  ObjectTable: ffff800ed4820c80  HandleCount: 3961.
    Image: System

3040: Object: ffffbe0c8cd66c30  GrantedAccess: 001f0006 Entry: ffff800edcbf7100
Object: ffffbe0c8cd66c30  Type: (ffffbe0c840fe7a0) File
    ObjectHeader: ffffbe0c8cd66c00 (new version)
        HandleCount: 1  PointerCount: 32768
        Directory Object: 00000000  Name: \Sessions\1\AppContainerNamedObjects\S-1-15-2-466767348-
3739614953-2700836392-1801644223-4227750657-1087833535-2488631167 {NamedPipe}

PROCESS ffffbe0c840eb040
    SessionId: none  Cid: 0004    Peb: 00000000  ParentCid: 0000
    DirBase: 001ae002  ObjectTable: ffff800ed4820c80  HandleCount: 3961.
    Image: System

308c: Object: ffffbe0c8b28f6d0  GrantedAccess: 001f0006 (Inherit) Entry: ffff800edcbf7230
Object: ffffbe0c8b28f6d0  Type: (ffffbe0c840fe7a0) File
    ObjectHeader: ffffbe0c8b28f6a0 (new version)
        HandleCount: 1  PointerCount: 32768
        Directory Object: 00000000  Name: \Sessions\1\AppContainerNamedObjects\S-1-15-2-1880626798-
2296700190-2192216202-2581987570-949377748-777141861-2889999867 {NamedPipe}

PROCESS ffffbe0c840eb040
    SessionId: none  Cid: 0004    Peb: 00000000  ParentCid: 0000
    DirBase: 001ae002  ObjectTable: ffff800ed4820c80  HandleCount: 3961.
    Image: System

3618: Object: ffffbe0c8cd5f700  GrantedAccess: 001f0006 (Protected) (Audit) Entry: ffff800edd6fe860
Object: ffffbe0c8cd5f700  Type: (ffffbe0c840fe7a0) File
    ObjectHeader: ffffbe0c8cd5f6d0 (new version)
        HandleCount: 1  PointerCount: 32768
        Directory Object: 00000000  Name: \Sessions\1\AppContainerNamedObjects\S-1-15-2-1050576210-
4101474698-56307613-2706264498-167457550-835605972-784472318 {NamedPipe}

[...]
```

11. Close the log file:

```
1: kd> .logclose
Closing open log file C:\AWMA-Dumps\M6.log
```



Memory Acquisition

https://www.patterndiagnostics.com/files/LegacyWindowsDebugging.pdf

Here I provide a link to a PDF file. Just look at the Special Topics slides.

https://www.patterndiagnostics.com/files/LegacyWindowsDebugging.pdf

Pattern Links

- Self-Diagnosis
- Driver Device Collection
- Raw Pointer
- Out-of-Module Pointer
- Deviant Token
- Hidden Process
- Stack Trace Collection (I/O)

Here are links to descriptions of patterns we found in our last 3 exercises (also available in Memory Dump Analysis Anthology, Encyclopedia of Crash Dump Analysis Patterns, and in this book Appendix):

Self-Diagnosis
https://www.dumpanalysis.org/blog/index.php/2011/04/26/crash-dump-analysis-patterns-part-69b/

Driver Device Collection
https://www.dumpanalysis.org/blog/index.php/2013/01/20/malware-analysis-patterns-part-10/

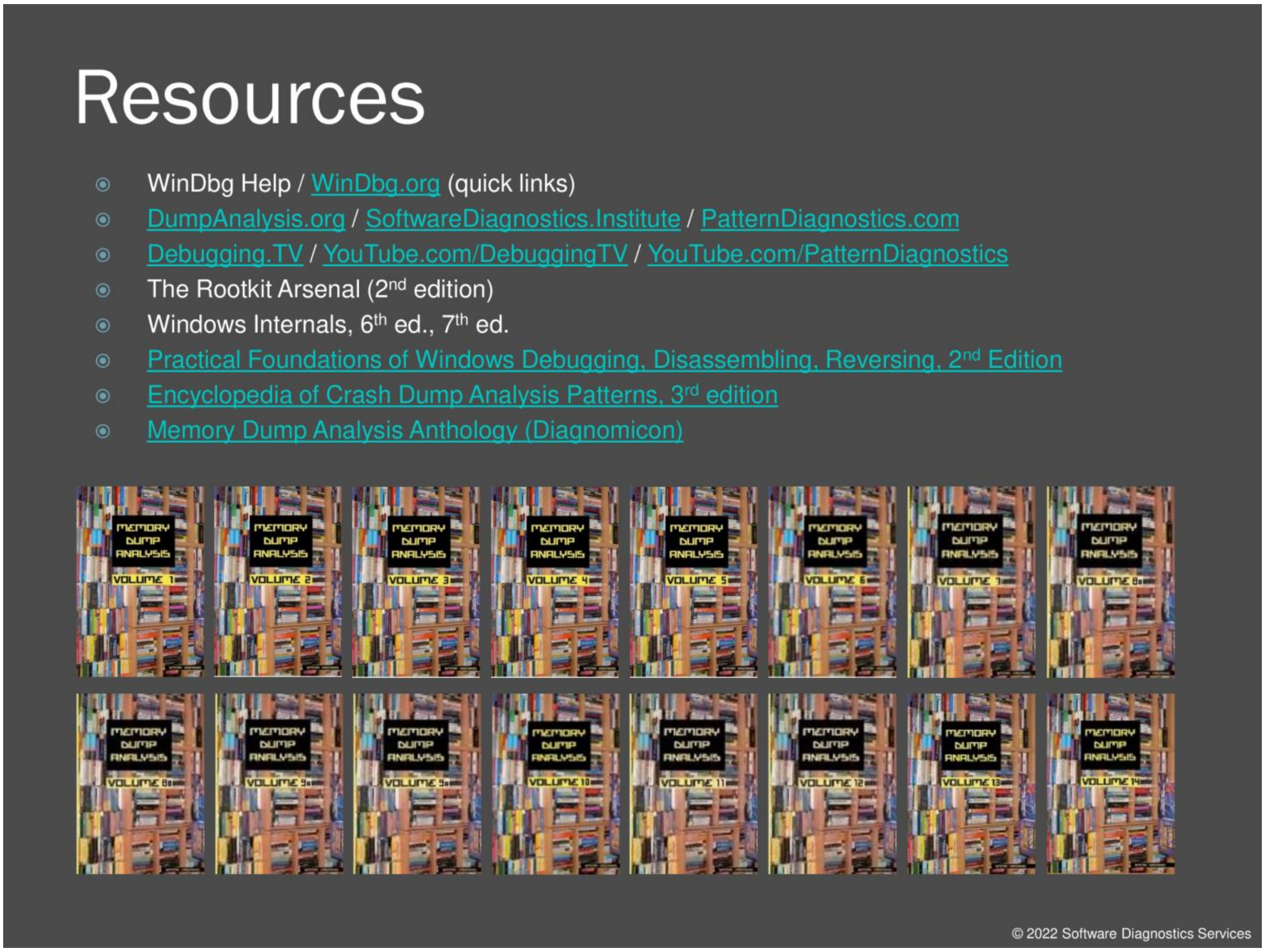
Raw Pointer
https://www.dumpanalysis.org/blog/index.php/2013/02/09/malware-analysis-patterns-part-22/

Out-of-Module Pointer
https://www.dumpanalysis.org/blog/index.php/2013/02/10/malware-analysis-patterns-part-23/

Deviant Token
https://www.dumpanalysis.org/blog/index.php/2012/12/31/crash-dump-analysis-patterns-part-191/

Hidden Process
https://www.dumpanalysis.org/blog/index.php/2012/11/13/crash-dump-analysis-patterns-part-186/

Stack Trace Collection (I/O)
https://www.dumpanalysis.org/blog/index.php/2012/01/11/crash-dump-analysis-patterns-part-27d/



Resources

- WinDbg Help / WinDbg.org (quick links)
- DumpAnalysis.org / SoftwareDiagnostics.Institute / PatternDiagnostics.com
- Debugging.TV / YouTube.com/DebuggingTV / YouTube.com/PatternDiagnostics
- The Rootkit Arsenal (2nd edition)
- Windows Internals, 6th ed., 7th ed.
- Practical Foundations of Windows Debugging, Disassembling, Reversing, 2nd Edition
- Encyclopedia of Crash Dump Analysis Patterns, 3rd edition
- Memory Dump Analysis Anthology (Diagnomicon)

MEMORY DUMP ANALYSIS VOLUME 1
MEMORY DUMP ANALYSIS VOLUME 2
MEMORY DUMP ANALYSIS VOLUME 3
MEMORY DUMP ANALYSIS VOLUME 4
MEMORY DUMP ANALYSIS VOLUME 5
MEMORY DUMP ANALYSIS VOLUME 6
MEMORY DUMP ANALYSIS VOLUME 7
MEMORY DUMP ANALYSIS VOLUME 8a
MEMORY DUMP ANALYSIS VOLUME 8b
MEMORY DUMP ANALYSIS VOLUME 9a
MEMORY DUMP ANALYSIS VOLUME 9b
MEMORY DUMP ANALYSIS VOLUME 10
MEMORY DUMP ANALYSIS VOLUME 11
MEMORY DUMP ANALYSIS VOLUME 12
MEMORY DUMP ANALYSIS VOLUME 13
MEMORY DUMP ANALYSIS VOLUME 14

A few notes about references. The Rootkit Arsenal book is very useful as it discusses the very opposite of what we were doing. If you need the basics of assembly language for 32-bit and 64-bit systems, such as function calls, their prologs and epilogs, and parameter passing, then you can find the Practical Foundations of Windows Debugging, Disassembling, and Reversing book useful.

Selected Q&A

Q. If you have a suspicious .dll file but not a memory dump, can you load the DLL directly into WinDbg?

A. Yes, it is the same as in Exercise M1A.

Q. What is the best way to take a crash dump? Task Manager vs. Process Explorer vs. WinDbg itself?

A. For running processes, the simplest way is to use Task Manager. You can use Task Manager even to trigger a kernel or complete memory dump by simply killing csrss.exe process.

Q. Back-tracking a pointer reference, what is a good pattern for that?

A. Please have a look at **Value References** pattern (also available in Volume 7 of Memory Dump Analysis Anthology and Encyclopedia of Crash Dump Analysis Patterns): https://www.dumpanalysis.org/blog/index.php/2011/12/05/crash-dump-analysis-patterns-part-159/.

Q. Is there a way to mark memory to be excluded from crashes? (eg. disk encryption keys, smb-hashes, etc...)

A. WinDbg has some limited capability here. Please check these posts (also available in Volumes 1 and 2 of Memory Dump Analysis Anthology): https://www.dumpanalysis.org/blog/index.php/2007/07/08/windbg-is-privacy-aware/ and https://www.dumpanalysis.org/blog/index.php/2008/09/09/beware-of-peb-data/.

For complete memory dumps, you can dump all processes and threads and other information into a textual log file and then inspect it for any sensitive information.

WinDbg also has scripting capability. Please look at a collection of scripts (available in various volumes of Memory Dump Analysis Anthology): https://www.dumpanalysis.org/blog/index.php/category/windbg-scripts/ and there is also a tutorial for C/C++ programmers: https://www.dumpanalysis.org/WCDA/WCDA-Sample-Chapter.pdf

Q. What does the "deferred" mean in the **lm** output?

A. It means that a symbol file for a module wasn't yet loaded because no addresses were found that need symbol mapping. But as soon as there is a pointer in that module address range (such as when using dps command), the corresponding PDF file is loaded.

Q. Is there anything we can do to include the paged out memory at the time of the crash dump?

A. When you save a full process memory dump, all paged out virtual process user space address range is brought from a page file. If you generate a complete memory dump to make sure that some processes of interest have all user space paged in, you can save their process memory dumps from Task Manager and then quickly force a complete memory dump.

Appendix

Malware Analysis Patterns

(reprinted with corrections from Memory Dump Analysis Anthology volumes and Encyclopedia of Crash Dump Analysis Patterns)

Deviant Module

When looking at the module list (**lmv**), searching for modules (**.imgscan),** or examining the particular module (**!address**, **!dh**), we may notice one of them as **deviant**. The deviation may be in (but not limited to as anything is possible):

- suspicious module name
- suspicious protection
- suspicious module load address

```
0:005> .imgscan
MZ at 00040000, prot 00000040, type 00020000 - size 1d000
MZ at 00340000, prot 00000002, type 01000000 - size 9c000
Name: iexplore.exe
MZ at 02250000, prot 00000002, type 00040000 - size 2000
MZ at 023b0000, prot 00000002, type 01000000 - size b000
Name: msimtf.dll
MZ at 03f80000, prot 00000002, type 00040000 - size 2000
MZ at 10000000, prot 00000004, type 00020000 - size 5000
Name: screens_dll.dll
MZ at 16080000, prot 00000002, type 01000000 - size 25000
Name: mdnsNSP.dll
MZ at 6ab50000, prot 00000002, type 01000000 - size 26000
Name: DSSENH.dll
MZ at 6b030000, prot 00000002, type 01000000 - size 5b0000
Name: MSHTML.dll
MZ at 6ba10000, prot 00000002, type 01000000 - size b4000
Name: JSCRIPT.dll
MZ at 6cec0000, prot 00000002, type 01000000 - size 1b000
Name: CRYPTNET.dll
MZ at 6d260000, prot 00000002, type 01000000 - size e000
Name: PNGFILTER.DLL
MZ at 6d2f0000, prot 00000002, type 01000000 - size 29000
Name: msls31.dll
MZ at 6d700000, prot 00000002, type 01000000 - size 30000
Name: MLANG.dll
MZ at 6d740000, prot 00000002, type 01000000 - size 4d000
Name: SSV.DLL
MZ at 6d7b0000, prot 00000002, type 01000000 - size c000
Name: ImgUtil.dll
MZ at 6ddb0000, prot 00000002, type 01000000 - size 2f000
Name: iepeers.DLL
MZ at 6df20000, prot 00000002, type 01000000 - size 33000
Name: IEShims.dll
MZ at 6eb80000, prot 00000002, type 01000000 - size a94000
Name: IEFRAME.dll
```

```
MZ at 703b0000, prot 00000002, type 01000000 - size 53000
Name: SWEEPRX.dll
MZ at 70740000, prot 00000002, type 01000000 - size 40000
Name: SWEEPRX.dll
MZ at 725a0000, prot 00000002, type 01000000 - size 12000
Name: PNRPNSP.dll
MZ at 725d0000, prot 00000002, type 01000000 - size 8000
Name: WINRNR.dll
MZ at 725e0000, prot 00000002, type 01000000 - size 136000
Name: MSXML3.dll
MZ at 72720000, prot 00000002, type 01000000 - size c000
Name: wshbth.dll
MZ at 72730000, prot 00000002, type 01000000 - size f000
Name: NAPINSP.dll
MZ at 72890000, prot 00000002, type 01000000 - size 6000
Name: SensApi.dll
MZ at 72ec0000, prot 00000002, type 01000000 - size 42000
Name: WINSPOOL.DRV
MZ at 734b0000, prot 00000002, type 01000000 - size 6000
Name: rasadhlp.dll
MZ at 736b0000, prot 00000002, type 01000000 - size 85000
Name: COMCTL32.dll
MZ at 73ac0000, prot 00000002, type 01000000 - size 7000
Name: MIDIMAP.dll
MZ at 73ae0000, prot 00000002, type 01000000 - size 14000
Name: MSACM32.dll
MZ at 73b00000, prot 00000002, type 01000000 - size 66000
Name: audioeng.dll
MZ at 73c30000, prot 00000002, type 01000000 - size 9000
Name: MSACM32.DRV
MZ at 73c60000, prot 00000002, type 01000000 - size 21000
Name: AudioSes.DLL
MZ at 73c90000, prot 00000002, type 01000000 - size 2f000
Name: WINMMDRV.dll
MZ at 74290000, prot 00000002, type 01000000 - size bb000
Name: PROPSYS.dll
MZ at 74390000, prot 00000002, type 01000000 - size f000
Name: nlaapi.dll
MZ at 743a0000, prot 00000002, type 01000000 - size 4000
Name: ksuser.dll
MZ at 74430000, prot 00000002, type 01000000 - size 15000
Name: Cabinet.dll
MZ at 74450000, prot 00000002, type 01000000 - size 3d000
Name: OLEACC.dll
MZ at 74490000, prot 00000002, type 01000000 - size 1ab000
Name: gdiplus.dll
MZ at 74640000, prot 00000002, type 01000000 - size 28000
Name: MMDevAPI.DLL
MZ at 74670000, prot 00000002, type 01000000 - size 32000
Name: WINMM.dll
MZ at 746b0000, prot 00000002, type 01000000 - size 31000
Name: TAPI32.dll
MZ at 749e0000, prot 00000002, type 01000000 - size 19e000
Name: COMCTL32.dll
```

```
MZ at 74b80000, prot 00000002, type 01000000 - size 7000
Name: AVRT.dll
MZ at 74ba0000, prot 00000002, type 01000000 - size 4a000
Name: RASAPI32.dll
MZ at 74ce0000, prot 00000002, type 01000000 - size 3f000
Name: UxTheme.dll
MZ at 74de0000, prot 00000002, type 01000000 - size 2d000
Name: WINTRUST.dll
MZ at 74ea0000, prot 00000002, type 01000000 - size 14000
Name: rasman.dll
MZ at 74f70000, prot 00000002, type 01000000 - size c000
Name: rtutils.dll
MZ at 74f80000, prot 00000002, type 01000000 - size 5000
Name: WSHTCPIP.dll
MZ at 74fb0000, prot 00000002, type 01000000 - size 21000
Name: NTMARTA.dll
MZ at 75010000, prot 00000002, type 01000000 - size 3b000
Name: RSAENH.dll
MZ at 75050000, prot 00000002, type 01000000 - size 5000
Name: MSIMG32.dll
MZ at 75060000, prot 00000002, type 01000000 - size 15000
Name: GPAPI.dll
MZ at 750a0000, prot 00000002, type 01000000 - size 46000
Name: SCHANNEL.dll
MZ at 752b0000, prot 00000002, type 01000000 - size 3b000
Name: MSWSOCK.dll
MZ at 75370000, prot 00000002, type 01000000 - size 45000
Name: bcrypt.dll
MZ at 753f0000, prot 00000002, type 01000000 - size 5000
Name: WSHIP6.dll
MZ at 75400000, prot 00000002, type 01000000 - size 8000
Name: VERSION.dll
MZ at 75420000, prot 00000002, type 01000000 - size 7000
Name: CREDSSP.dll
MZ at 75430000, prot 00000002, type 01000000 - size 35000
Name: ncrypt.dll
MZ at 75480000, prot 00000002, type 01000000 - size 22000
Name: dhcpcsvc6.DLL
MZ at 754b0000, prot 00000002, type 01000000 - size 7000
Name: WINNSI.DLL
MZ at 754c0000, prot 00000002, type 01000000 - size 35000
Name: dhcpcsvc.DLL
MZ at 75500000, prot 00000002, type 01000000 - size 19000
Name: IPHLPAPI.DLL
MZ at 75590000, prot 00000002, type 01000000 - size 3a000
Name: slc.dll
MZ at 755d0000, prot 00000002, type 01000000 - size f2000
Name: CRYPT32.dll
MZ at 75740000, prot 00000002, type 01000000 - size 12000
Name: MSASN1.dll
MZ at 75760000, prot 00000002, type 01000000 - size 11000
Name: SAMLIB.dll
MZ at 75780000, prot 00000002, type 01000000 - size 76000
Name: NETAPI32.dll
```

```
MZ at 75800000, prot 00000002, type 01000000 - size 2c000
Name: DNSAPI.dll
MZ at 75a70000, prot 00000002, type 01000000 - size 5f000
Name: sxs.dll
MZ at 75ad0000, prot 00000002, type 01000000 - size 2c000
Name: apphelp.dll
MZ at 75b30000, prot 00000002, type 01000000 - size 14000
Name: Secur32.dll
MZ at 75b50000, prot 00000002, type 01000000 - size 1e000
Name: USERENV.dll
MZ at 75c90000, prot 00000002, type 01000000 - size 7000
Name: PSAPI.DLL
MZ at 75ca0000, prot 00000002, type 01000000 - size c3000
Name: RPCRT4.dll
MZ at 75d70000, prot 00000002, type 01000000 - size 73000
Name: COMDLG32.dll
MZ at 75df0000, prot 00000002, type 01000000 - size 9000
Name: LPK.dll
MZ at 75e00000, prot 00000002, type 01000000 - size dc000
Name: KERNEL32.dll
MZ at 75ee0000, prot 00000002, type 01000000 - size aa000
Name: msvcrt.dll
MZ at 75f90000, prot 00000002, type 01000000 - size 1e8000
Name: iertutil.dll
MZ at 76180000, prot 00000002, type 01000000 - size 29000
Name: imagehlp.dll
MZ at 761b0000, prot 00000002, type 01000000 - size 6000
Name: NSI.dll
MZ at 761c0000, prot 00000002, type 01000000 - size 84000
Name: CLBCatQ.DLL
MZ at 76250000, prot 00000002, type 01000000 - size 49000
Name: WLDAP32.dll
MZ at 762a0000, prot 00000002, type 01000000 - size c6000
Name: ADVAPI32.dll
MZ at 76370000, prot 00000002, type 01000000 - size 4b000
Name: GDI32.dll
MZ at 763c0000, prot 00000002, type 01000000 - size 59000
Name: SHLWAPI.dll
MZ at 76420000, prot 00000002, type 01000000 - size e6000
Name: WININET.dll
MZ at 76510000, prot 00000002, type 01000000 - size b10000
Name: SHELL32.dll
MZ at 77020000, prot 00000002, type 01000000 - size 145000
Name: ole32.dll
MZ at 77170000, prot 00000002, type 01000000 - size 7d000
Name: USP10.dll
MZ at 771f0000, prot 00000002, type 01000000 - size 8d000
Name: OLEAUT32.dll
MZ at 77280000, prot 00000002, type 01000000 - size 18a000
Name: SETUPAPI.dll
MZ at 77410000, prot 00000002, type 01000000 - size 9d000
Name: USER32.dll
MZ at 774b0000, prot 00000002, type 01000000 - size 133000
Name: urlmon.dll
```

```
MZ at 775f0000, prot 00000002, type 01000000 - size 127000
Name: ntdll.dll
MZ at 77720000, prot 00000002, type 01000000 - size 3000
Name: Normaliz.dll
MZ at 77730000, prot 00000002, type 01000000 - size 2d000
Name: WS2_32.dll
MZ at 77760000, prot 00000002, type 01000000 - size 1e000
Name: IMM32.dll
MZ at 77780000, prot 00000002, type 01000000 - size c8000
Name: MSCTF.dll
MZ at 7c340000, prot 00000002, type 01000000 - size 56000
Name: MSVCR71.dll

0:005> !address 00040000
Usage:                  <unclassified>
Allocation Base:        00040000
Base Address:           00040000
End Address:            0005d000
Region Size:            0001d000
Type:                   00020000 MEM_PRIVATE
State:                  00001000 MEM_COMMIT
Protect:                00000040 PAGE_EXECUTE_READWRITE

0:005> !address 10000000
Usage:                  <unclassified>
Allocation Base:        10000000
Base Address:           10000000
End Address:            10001000
Region Size:            00001000
Type:                   00020000 MEM_PRIVATE
State:                  00001000 MEM_COMMIT
Protect:                00000004 PAGE_READWRITE
```

- suspicious text inside

  See Volume 5, page 406 for a case study example.

- suspicious import table (for example, screen grabbing) or its absence (dynamic imports)

```
0:005> !dh 10000000
[...]
2330 [      50] address [size] of Export Directory
20E0 [      78] address [size] of Import Directory
0 [       0] address [size] of Resource Directory
0 [       0] address [size] of Exception Directory
0 [       0] address [size] of Security Directory
4000 [      34] address [size] of Base Relocation Directory
2060 [      1C] address [size] of Debug Directory
0 [       0] address [size] of Description Directory
0 [       0] address [size] of Special Directory
0 [       0] address [size] of Thread Storage Directory
0 [       0] address [size] of Load Configuration Directory
0 [       0] address [size] of Bound Import Directory
```

```
2000 [      58] address [size] of Import Address Table Directory
0 [       0] address [size] of Delay Import Directory
0 [       0] address [size] of COR20 Header Directory
0 [       0] address [size] of Reserved Directory
[...]

0:005> dps 10000000+2000 10000000+2000+58
10002000 76376101 gdi32!CreateCompatibleDC
10002004 763793d6 gdi32!StretchBlt
10002008 76377461 gdi32!CreateDIBSection
1000200c 763762a0 gdi32!SelectObject
10002010 00000000
10002014 75e4a411 kernel32!lstrcmpW
10002018 75e440aa kernel32!VirtualFree
1000201c 75e4ad55 kernel32!VirtualAlloc
10002020 00000000
10002024 77429ced user32!ReleaseDC
10002028 77423ba7 user32!NtUserGetWindowDC
1000202c 77430e21 user32!GetWindowRect
10002030 00000000
10002034 744a75e9 GdiPlus!GdiplusStartup
10002038 744976dd GdiPlus!GdipSaveImageToStream
1000203c 744cdd38 GdiPlus!GdipGetImageEncodersSize
10002040 744971cf GdiPlus!GdipDisposeImage
10002044 744a8591 GdiPlus!GdipCreateBitmapFromHBITMAP
10002048 744cdbae GdiPlus!GdipGetImageEncoders
1000204c 00000000
10002050 7707d51b ole32!CreateStreamOnHGlobal
10002054 00000000
10002058 00000000

0:000> !dh 012a0000
[...]
0 [       0] address [size] of Export Directory
0 [       0] address [size] of Import Directory
0 [       0] address [size] of Resource Directory
0 [       0] address [size] of Exception Directory
0 [       0] address [size] of Security Directory
8000 [      FC] address [size] of Base Relocation Directory
4000 [      1C] address [size] of Debug Directory
0 [       0] address [size] of Description Directory
0 [       0] address [size] of Special Directory
0 [       0] address [size] of Thread Storage Directory
0 [       0] address [size] of Load Configuration Directory
0 [       0] address [size] of Bound Import Directory
0 [       0] address [size] of Import Address Table Directory
0 [       0] address [size] of Delay Import Directory
0 [       0] address [size] of COR20 Header Directory
0 [       0] address [size] of Reserved Directory
[...]
```

- suspicious path names

```
Age: 7, Pdb: d:\work\BekConnekt\Client_src_code_New\Release\Blackjoe_new.pdb

Debug Directories(1)
Type Size Address Pointer
cv 46 2094 894 Format: RSDS, guid, 1, C:\MyWork\screens_dll\Release\screens_dll.pdb
```

- suspicious image path (although it could be dynamic code generation for .NET assemblies)
- uninitialized image resources

```
0:002> lmv m C6DC
start    end        module name
012a0000 012a9000   C6DC     C (no symbols)
Loaded symbol image file: C6DC.tmp
Image path: C:\Users\User\AppData\Local\Temp\C6DC.tmp
Image name: C6DC.tmp
Timestamp:        Sun May 30 20:18:32 2010 (4C02BA08)
CheckSum:         00000000
ImageSize:        00009000
File version:     0.0.0.0
Product version:  0.0.0.0
File flags:       0 (Mask 0)
File OS:          0 Unknown Base
File type:        0.0 Unknown
File date:        00000000.00000000
Translations:     0000.04b0 0000.04e4 0409.04b0 0409.04e4
```

Deviant Token

Sometimes we need to check under what security principal or group we run a process or what privileges it has, or whether it has impersonating threads. We may find an unexpected token with a different security identifier, for example, Network Service instead of Local System (SID: S-1-5-18):

```
PROCESS 8f218d88  SessionId: 0  Cid: 09c4    Peb: 7ffdf000  ParentCid: 0240
DirBase: bffd4260  ObjectTable: e10eae90  HandleCount:  93.
Image: ServiceA.exe
VadRoot 8f1f70e8 Vads 141 Clone 0 Private 477. Modified 2. Locked 0.
DeviceMap e10038d8
Token                             e10ff5d8
[...]

0: kd> !token e10ff5d8
_TOKEN e10ff5d8
TS Session ID: 0
User: S-1-5-20
[...]
```

Well-known SIDs can be found in this MS article: https://docs.microsoft.com/en-GB/windows/security/identity-protection/access-control/security-identifiers.

Driver Device Collection

This pattern can be used to compare the current list of device and driver objects with some saved reference list to find out any changes. This listing can be done by using **!object** command:

```
0: kd> !object \Driver
[...]

0: kd> !object \FileSystem
[...]

0: kd> !object \Device
[...]
```

Note that the collection is called **Driver Device** and not Device Driver.

Execution Residue

For the pattern about NULL code pointer (Volume 2, page 237), I created a simple program that crashes when we pass a NULL thread procedure pointer to *CreateThread* function. We might expect to see little in the raw stack data (Volume 1, page 231) because there was no user-supplied thread code. In reality, if we dump it, we would see lots of symbolic information for code and data, including ASCII and UNICODE fragments that I call **Execution Residue** patterns, and one of them is **Exception Handling Residue** we can use to check for **Hidden Exceptions** (Volume 1, page 271) and differentiate between 1st and 2nd chance exceptions (Volume 1, page 109). Code residues are very powerful in reconstructing stack traces manually (Volume 1, page 157) or looking for partial stack traces and historical information (Volume 1, page 457).

To show typical execution residues, I created another small program with two additional threads based on the Visual Studio Win32 project. After we dismiss the About box, we create the first thread, and then we crash the process when creating the second thread because of the NULL thread procedure:

```
typedef DWORD (WINAPI *THREADPROC)(PVOID);

DWORD WINAPI ThreadProc(PVOID pvParam)
{
   for (unsigned int i = 0xFFFFFFFF; i; --i);
   return 0;
}

// Message handler for about box.
INT_PTR CALLBACK About(HWND hDlg, UINT message, WPARAM wParam, LPARAM lParam)
{
   UNREFERENCED_PARAMETER(lParam);
   switch (message)
   {
   case WM_INITDIALOG:
      return (INT_PTR)TRUE;

   case WM_COMMAND:
      if (LOWORD(wParam) == IDOK || LOWORD(wParam) == IDCANCEL)
      {
         EndDialog(hDlg, LOWORD(wParam));
         THREADPROC thProc = ThreadProc;
         HANDLE hThread = CreateThread(NULL, 0, ThreadProc, 0, 0, NULL);
         CloseHandle(hThread);
         Sleep(1000);
         hThread = CreateThread(NULL, 0, NULL, 0, 0, NULL);
         CloseHandle(hThread);
         return (INT_PTR)TRUE;
      }
      break;
   }
   return (INT_PTR)FALSE;
}
```

When we open the crash dump we see these threads:

```
0:002> ~*kL

   0  Id: cb0.9ac Suspend: 1 Teb: 7efdd000 Unfrozen
ChildEBP RetAddr
0012fdf4 00411554 user32!NtUserGetMessage+0x15
0012ff08 00412329 NullThread!wWinMain+0xa4
0012ffb8 0041208d NullThread!__tmainCRTStartup+0x289
```

```
0012ffc0 7d4e7d2a NullThread!wWinMainCRTStartup+0xd
0012fff0 00000000 kernel32!BaseProcessStart+0x28

   1  Id: cb0.8b4 Suspend: 1 Teb: 7efda000 Unfrozen
ChildEBP RetAddr
01eafea4 7d63f501 ntdll!NtWaitForMultipleObjects+0x15
01eaff48 7d63f988 ntdll!EtwpWaitForMultipleObjectsEx+0xf7
01eaffb8 7d4dfe21 ntdll!EtwpEventPump+0x27f
01eaffec 00000000 kernel32!BaseThreadStart+0x34

   2  Id: cb0.ca8 Suspend: 1 Teb: 7efd7000 Unfrozen
ChildEBP RetAddr
0222ffb8 7d4dfe21 NullThread!ThreadProc+0x34
0222ffec 00000000 kernel32!BaseThreadStart+0x34

#  3  Id: cb0.5bc Suspend: 1 Teb: 7efaf000 Unfrozen
ChildEBP RetAddr
WARNING: Frame IP not in any known module. Following frames may be wrong.
0236ffb8 7d4dfe21 0x0
0236ffec 00000000 kernel32!BaseThreadStart+0x34

   4  Id: cb0.468 Suspend: -1 Teb: 7efac000 Unfrozen
ChildEBP RetAddr
01f7ffb4 7d674807 ntdll!NtTerminateThread+0x12
01f7ffc4 7d66509f ntdll!RtlExitUserThread+0x26
01f7fff4 00000000 ntdll!DbgUiRemoteBreakin+0x41
```

We see our first created thread looping:

```
0:003> ~2s
eax=cbcf04b5 ebx=00000000 ecx=00000000 edx=00000000 esi=00000000 edi=0222ffb8
eip=00411aa4 esp=0222fee0 ebp=0222ffb8 iopl=0 nv up ei ng nz na po nc
cs=0023 ss=002b ds=002b es=002b fs=0053 gs=002b efl=00000282
NullThread!ThreadProc+0x34:
00411aa4 7402   je      NullThread!ThreadProc+0x38 (00411aa8)   [br=0]

0:002> u
NullThread!ThreadProc+0x34:
00411aa4 je      NullThread!ThreadProc+0x38 (00411aa8)
00411aa6 jmp     NullThread!ThreadProc+0x27 (00411a97)
00411aa8 xor     eax,eax
00411aaa pop     edi
00411aab pop     esi
00411aac pop     ebx
00411aad mov     esp,ebp
00411aaf pop     ebp
```

We might expect it to have very little in its raw stack data, but what we see when we dump its stack range from **!teb** command is **Thread Startup Residue,** where some symbolic information might be coincidental too (Volume 1, page 390):

```
0:002> dds 0222f000  02230000
0222f000  00000000
0222f004  00000000
0222f008  00000000
...
0222f104  00000000
0222f108  00000000
0222f10c  00000000
0222f110  7d621954 ntdll!RtlImageNtHeaderEx+0xee
0222f114  7efde000
```

```
0222f118  00000000
0222f11c  00000001
0222f120  000000e8
0222f124  004000e8 NullThread!_enc$textbss$begin <PERF> (NullThread+0xe8)
0222f128  00000000
0222f12c  0222f114
0222f130  00000000
0222f134  0222fca0
0222f138  7d61f1f8 ntdll!_except_handler3
0222f13c  7d621958 ntdll!RtlpRunTable+0x4a0
0222f140  ffffffff
0222f144  7d621954 ntdll!RtlImageNtHeaderEx+0xee
0222f148  7d6218ab ntdll!RtlImageNtHeader+0x1b
0222f14c  00000001
0222f150  00400000 NullThread!_enc$textbss$begin <PERF> (NullThread+0x0)
0222f154  00000000
0222f158  00000000
0222f15c  0222f160
0222f160  004000e8 NullThread!_enc$textbss$begin <PERF> (NullThread+0xe8)
0222f164  0222f7bc
0222f168  7d4dfea3 kernel32!ConsoleApp+0xe
0222f16c  00400000 NullThread!_enc$textbss$begin <PERF> (NullThread+0x0)
0222f170  7d4dfe77 kernel32!ConDllInitialize+0x1f5
0222f174  00000000
0222f178  7d4dfe8c kernel32!ConDllInitialize+0x20a
0222f17c  00000000
0222f180  00000000
...
0222f290  00000000
0222f294  0222f2b0
0222f298  7d6256e8 ntdll!bsearch+0x42
0222f29c  00180144
0222f2a0  0222f2b4
0222f2a4  7d625992 ntdll!ARRAY_FITS+0x29
0222f2a8  00000a8c
0222f2ac  00001f1c
0222f2b0  0222f2c0
0222f2b4  0222f2f4
0222f2b8  7d625944 ntdll!RtlpLocateActivationContextSection+0x1da
0222f2bc  00001f1c
0222f2c0  000029a8
...
0222f2e0  536cd652
0222f2e4  0222f334
0222f2e8  7d625b62 ntdll!RtlpFindUnicodeStringInSection+0x7b
0222f2ec  0222f418
0222f2f0  00000000
0222f2f4  0222f324
0222f2f8  7d6257f1 ntdll!RtlpFindNextActivationContextSection+0x64
0222f2fc  00181f1c
0222f300  c0150008
...
0222f320  7efd7000
0222f324  0222f344
0222f328  7d625cd2 ntdll!RtlFindNextActivationContextSection+0x46
0222f32c  0222f368
0222f330  0222f3a0
0222f334  0222f38c
0222f338  0222f340
0222f33c  00181f1c
0222f340  00000000
0222f344  0222f390
```

```
0222f348  7d625ad8 ntdll!RtlFindActivationContextSectionString+0xe1
0222f34c  0222f368
0222f350  0222f3a0
...
0222f38c  00000a8c
0222f390  0222f454
0222f394  7d626381 ntdll!CsrCaptureMessageMultiUnicodeStringsInPlace+0xa57
0222f398  00000003
0222f39c  00000000
0222f3a0  00181f1c
0222f3a4  0222f418
0222f3a8  0222f3b4
0222f3ac  7d6a0340 ntdll!LdrApiDefaultExtension
0222f3b0  7d6263df ntdll!CsrCaptureMessageMultiUnicodeStringsInPlace+0xb73
0222f3b4  00000040
0222f3b8  00000000
...
0222f420  00000000
0222f424  0222f458
0222f428  7d625f9a ntdll!CsrCaptureMessageMultiUnicodeStringsInPlace+0x4c1
0222f42c  00020000
0222f430  0222f44c
0222f434  0222f44c
0222f438  0222f44c
0222f43c  00000002
0222f440  00000002
0222f444  7d625f9a ntdll!CsrCaptureMessageMultiUnicodeStringsInPlace+0x4c1
0222f448  00020000
0222f44c  00000000
0222f450  00003cfb
0222f454  0222f5bc
0222f458  0222f4f4
0222f45c  0222f5bc
0222f460  7d626290 ntdll!RtlDosApplyFileIsolationRedirection_Ustr+0x346
0222f464  0222f490
0222f468  00000000
0222f46c  0222f69c
0222f470  7d6262f5 ntdll!RtlDosApplyFileIsolationRedirection_Ustr+0x3de
0222f474  0222f510
0222f478  7d6a0340 ntdll!LdrApiDefaultExtension
0222f47c  7d626290 ntdll!RtlDosApplyFileIsolationRedirection_Ustr+0x346
0222f480  00000000
0222f484  00800000
...
0222f544  00000000
0222f548  00000001
0222f54c  7d6a0290 ntdll!LdrpHashTable+0x50
0222f550  00000000
0222f554  00500000
...
0222f59c  00000000
0222f5a0  0222f5d4
0222f5a4  7d6251d0 ntdll!LdrUnlockLoaderLock+0x84
0222f5a8  7d6251d7 ntdll!LdrUnlockLoaderLock+0xad
0222f5ac  00000000
0222f5b0  0222f69c
0222f5b4  00000000
0222f5b8  00003cfb
0222f5bc  0222f5ac
0222f5c0  7d626de0 ntdll!LdrGetDllHandleEx+0xbe
0222f5c4  0222f640
0222f5c8  7d61f1f8 ntdll!_except_handler3
```

```
0222f5cc  7d6251e0 ntdll!`string'+0x74
0222f5d0  ffffffff
0222f5d4  7d6251d7 ntdll!LdrUnlockLoaderLock+0xad
0222f5d8  7d626fb3 ntdll!LdrGetDllHandleEx+0x368
0222f5dc  00000001
0222f5e0  0ca80042
0222f5e4  7d626f76 ntdll!LdrGetDllHandleEx+0x329
0222f5e8  00000000
0222f5ec  7d626d0b ntdll!LdrGetDllHandle
0222f5f0  00000002
0222f5f4  001a0018
...
0222f640  0222f6a8
0222f644  7d61f1f8 ntdll!_except_handler3
0222f648  7d626e60 ntdll!`string'+0xb4
0222f64c  ffffffff
0222f650  7d626f76 ntdll!LdrGetDllHandleEx+0x329
0222f654  7d626d23 ntdll!LdrGetDllHandle+0x18
0222f658  00000001
...
0222f66c  0222f6b8
0222f670  7d4dff0e kernel32!GetModuleHandleForUnicodeString+0x20
0222f674  00000001
0222f678  00000000
0222f67c  0222f6d4
0222f680  7d4dff1e kernel32!GetModuleHandleForUnicodeString+0x97
0222f684  00000000
0222f688  7efd7c00
0222f68c  00000002
0222f690  00000001
0222f694  00000000
0222f698  0222f6f0
0222f69c  7d4c0000 kernel32!_imp__NtFsControlFile <PERF> (kernel32+0x0)
0222f6a0  0222f684
0222f6a4  7efd7c00
0222f6a8  0222fb20
0222f6ac  7d4d89c4 kernel32!_except_handler3
0222f6b0  7d4dff28 kernel32!`string'+0x18
0222f6b4  ffffffff
0222f6b8  7d4dff1e kernel32!GetModuleHandleForUnicodeString+0x97
0222f6bc  7d4e001f kernel32!BasepGetModuleHandleExW+0x17f
0222f6c0  7d4e009f kernel32!BasepGetModuleHandleExW+0x23c
0222f6c4  00000000
0222f6c8  0222fc08
0222f6cc  00000001
0222f6d0  ffffffff
0222f6d4  001a0018
0222f6d8  7efd7c00
0222f6dc  0222fb50
0222f6e0  00000000
0222f6e4  00000000
0222f6e8  00000000
0222f6ec  02080000 oleaut32!_PictSaveEnhMetaFile+0x76
0222f6f0  0222f90c
0222f6f4  02080000 oleaut32!_PictSaveEnhMetaFile+0x76
0222f6f8  0222f704
0222f6fc  00000000
0222f700  7d4c0000 kernel32!_imp__NtFsControlFile <PERF> (kernel32+0x0)
0222f704  00000000
0222f708  02080000 oleaut32!_PictSaveEnhMetaFile+0x76
0222f70c  0222f928
0222f710  02080000 oleaut32!_PictSaveEnhMetaFile+0x76
```

```
0222f714  0222f720
0222f718  00000000
0222f71c  7d4c0000 kernel32!_imp__NtFsControlFile <PERF> (kernel32+0x0)
0222f720  00000000
0222f724  00000000
...
0222f7b8  0000f949
0222f7bc  0222fbf4
0222f7c0  7d4dfdd0 kernel32!_BaseDllInitialize+0x6b
0222f7c4  00000002
0222f7c8  00000000
0222f7cc  00000000
0222f7d0  7d4dfde4 kernel32!_BaseDllInitialize+0x495
0222f7d4  00000000
0222f7d8  7efde000
0222f7dc  7d4c0000 kernel32!_imp__NtFsControlFile <PERF> (kernel32+0x0)
0222f7e0  00000000
0222f7e4  00000000
...
0222f894  01c58ae0
0222f898  0222fac0
0222f89c  7d62155b ntdll!RtlAllocateHeap+0x460
0222f8a0  7d61f78c ntdll!RtlAllocateHeap+0xee7
0222f8a4  00000000
0222f8a8  0222fc08
...
0222f8d8  00000000
0222f8dc  7d621954 ntdll!RtlImageNtHeaderEx+0xee
0222f8e0  0222f9a4
0222f8e4  7d614c88 ntdll!$$VProc_ImageExportDirectory+0x2c48
0222f8e8  0222f9a6
0222f8ec  7d612040 ntdll!$$VProc_ImageExportDirectory
0222f8f0  00000221
0222f8f4  0222f944
0222f8f8  7d627405 ntdll!LdrpSnapThunk+0xc0
0222f8fc  0222f9a6
0222f900  00000584
0222f904  7d600000 ntdll!RtlDosPathSeperatorsString <PERF> (ntdll+0x0)
0222f908  7d613678 ntdll!$$VProc_ImageExportDirectory+0x1638
0222f90c  7d614c88 ntdll!$$VProc_ImageExportDirectory+0x2c48
0222f910  0222f9a4
0222f914  00000001
0222f918  0222f9a4
0222f91c  00000000
0222f920  0222f990
0222f924  7d6000f0 ntdll!RtlDosPathSeperatorsString <PERF> (ntdll+0xf0)
0222f928  0222f968
0222f92c  00000001
0222f930  0222f9a4
0222f934  7d6000f0 ntdll!RtlDosPathSeperatorsString <PERF> (ntdll+0xf0)
0222f938  0222f954
0222f93c  00000000
0222f940  00000000
0222f944  0222fa00
0222f948  7d62757a ntdll!LdrpGetProcedureAddress+0x189
0222f94c  0222f95c
0222f950  00000098
0222f954  00000005
0222f958  01c44f48
0222f95c  0222fb84
0222f960  7d62155b ntdll!RtlAllocateHeap+0x460
0222f964  7d61f78c ntdll!RtlAllocateHeap+0xee7
```

```
0222f968  00000000
0222f96c  0000008c
0222f970  00000000
0222f974  7d4d8472 kernel32!$$VProc_ImageExportDirectory+0x6d4e
0222f978  0222fa1c
0222f97c  7d627607 ntdll!LdrpGetProcedureAddress+0x274
0222f980  7d612040 ntdll!$$VProc_ImageExportDirectory
0222f984  002324f8
0222f988  7d600000 ntdll!RtlDosPathSeperatorsString <PERF> (ntdll+0x0)
0222f98c  0222faa8
0222f990  0000a7bb
0222f994  00221f08
0222f998  0222f9a4
0222f99c  7d627c2e ntdll!RtlDecodePointer
0222f9a0  00000000
0222f9a4  74520000
0222f9a8  6365446c
0222f9ac  5065646f
0222f9b0  746e696f
0222f9b4  00007265
0222f9b8  7d627c2e ntdll!RtlDecodePointer
0222f9bc  00000000
...
0222f9f8  01c40640
0222f9fc  00000000
0222fa00  7d6275b2 ntdll!LdrpGetProcedureAddress+0xb3
0222fa04  7d627772 ntdll!LdrpSnapThunk+0x31c
0222fa08  7d600000 ntdll!RtlDosPathSeperatorsString <PERF> (ntdll+0x0)
0222fa0c  0222fa44
0222fa10  00000000
0222fa14  0222faa8
0222fa18  00000000
0222fa1c  0222fab0
0222fa20  00000001
0222fa24  00000001
0222fa28  00000000
0222fa2c  0222fa9c
0222fa30  7d4c00e8 kernel32!_imp__NtFsControlFile <PERF> (kernel32+0xe8)
0222fa34  01c44fe0
0222fa38  00000001
0222fa3c  01c401a0
0222fa40  7d4c00e8 kernel32!_imp__NtFsControlFile <PERF> (kernel32+0xe8)
0222fa44  00110010
0222fa48  7d4d8478 kernel32!$$VProc_ImageExportDirectory+0x6d54
0222fa4c  00000000
0222fa50  0222fb0c
0222fa54  7d62757a ntdll!LdrpGetProcedureAddress+0x189
0222fa58  7d600000 ntdll!RtlDosPathSeperatorsString <PERF> (ntdll+0x0)
0222fa5c  00000000
0222fa60  0022faa8
0222fa64  0222fab0
0222fa68  0222fb0c
0222fa6c  7d627607 ntdll!LdrpGetProcedureAddress+0x274
0222fa70  7d6a0180 ntdll!LdrpLoaderLock
0222fa74  7d6275b2 ntdll!LdrpGetProcedureAddress+0xb3
0222fa78  102ce1ac msvcr80d!`string'
0222fa7c  0222fc08
0222fa80  0000ffff
0222fa84  0022f8b0
0222fa88  0022f8a0
0222fa8c  00000003
0222fa90  0222fbd4
```

```
0222fa94  020215fc oleaut32!DllMain+0x2c
0222fa98  02020000 oleaut32!_imp__RegFlushKey <PERF> (oleaut32+0x0)
0222fa9c  00000002
0222faa0  00000000
0222faa4  00000000
0222faa8  00000002
0222faac  0202162d oleaut32!DllMain+0x203
0222fab0  65440000
0222fab4  02020000 oleaut32!_imp__RegFlushKey <PERF> (oleaut32+0x0)
0222fab8  00000001
0222fabc  00726574
0222fac0  0222facc
0222fac4  7d627c2e ntdll!RtlDecodePointer
0222fac8  00000000
0222facc  65440000
0222fad0  00000000
0222fad4  00000000
0222fad8  00726574
0222fadc  00000005
0222fae0  00000000
0222fae4  1021af95 msvcr80d!_heap_alloc_dbg+0x375
0222fae8  002322f0
0222faec  00000000
0222faf0  01c40238
0222faf4  0222fa78
0222faf8  7efd7bf8
0222fafc  00000020
0222fb00  7d61f1f8 ntdll!_except_handler3
0222fb04  7d6275b8 ntdll!`string'+0xc
0222fb08  ffffffff
0222fb0c  7d6275b2 ntdll!LdrpGetProcedureAddress+0xb3
0222fb10  00000000
0222fb14  00000000
0222fb18  0222fb48
0222fb1c  00000000
0222fb20  01000000
0222fb24  00000001
0222fb28  0222fb50
0222fb2c  7d4dac3a kernel32!GetProcAddress+0x44
0222fb30  0222fb50
0222fb34  7d4dac4c kernel32!GetProcAddress+0x5c
0222fb38  0222fc08
0222fb3c  00000013
0222fb40  00000000
0222fb44  01c44f40
0222fb48  01c4015c
0222fb4c  00000098
0222fb50  01c44f40
0222fb54  01c44f48
0222fb58  01c40238
0222fb5c  10204f9f msvcr80d!_initptd+0x10f
0222fb60  00000098
0222fb64  00000000
0222fb68  01c40000
0222fb6c  0222f968
0222fb70  7d4c0000 kernel32!_imp__NtFsControlFile <PERF> (kernel32+0x0)
0222fb74  00000ca8
0222fb78  4b405064 msctf!g_timlist
0222fb7c  0222fbb8
0222fb80  4b3c384f msctf!CTimList::Leave+0x6
0222fb84  4b3c14d7 msctf!CTimList::IsThreadId+0x5a
0222fb88  00000ca8
```

```
0222fb8c  4b405064 msctf!g_timlist
0222fb90  4b3c0000 msctf!_imp__CheckTokenMembership <PERF> (msctf+0x0)
0222fb94  01c70000
0222fb98  00000000
0222fb9c  4b405064 msctf!g_timlist
0222fba0  0222fb88
0222fba4  7d4dfd40 kernel32!FlsSetValue+0xc7
0222fba8  0222fca0
0222fbac  4b401dbd msctf!_except_handler3
0222fbb0  4b3c14e0 msctf!`string'+0x78
0222fbb4  0222fbd4
0222fbb8  0022f8a0
0222fbbc  00000001
0222fbc0  00000000
0222fbc4  00000000
0222fbc8  0222fc80
0222fbcc  0022f8a0
0222fbd0  0000156f
0222fbd4  0222fbf4
0222fbd8  020215a4 oleaut32!_DllMainCRTStartup+0x52
0222fbdc  02020000 oleaut32!_imp__RegFlushKey <PERF> (oleaut32+0x0)
0222fbe0  00000002
0222fbe4  00000000
0222fbe8  00000000
0222fbec  0222fc08
0222fbf0  00000001
0222fbf4  0222fc14
0222fbf8  7d610024 ntdll!LdrpCallInitRoutine+0x14
0222fbfc  02020000 oleaut32!_imp__RegFlushKey <PERF> (oleaut32+0x0)
0222fc00  00000001
0222fc04  00000000
0222fc08  00000001
0222fc0c  00000000
0222fc10  0022f8a0
0222fc14  00000001
0222fc18  00000000
0222fc1c  0222fcb0
0222fc20  7d62822e ntdll!LdrpInitializeThread+0x1a5
0222fc24  7d6a0180 ntdll!LdrpLoaderLock
0222fc28  7d62821c ntdll!LdrpInitializeThread+0x18f
0222fc2c  00000000
0222fc30  7efde000
0222fc34  00000000
...
0222fc6c  00000070
0222fc70  ffffffff
0222fc74  ffffffff
0222fc78  7d6281c7 ntdll!LdrpInitializeThread+0xd8
0222fc7c  7d6280d6 ntdll!LdrpInitializeThread+0x12c
0222fc80  00000000
0222fc84  00000000
0222fc88  0022f8a0
0222fc8c  0202155c oleaut32!_DllMainCRTStartup
0222fc90  7efde000
0222fc94  7d6a01f4 ntdll!PebLdr+0x14
0222fc98  0222fc2c
0222fc9c  00000000
0222fca0  0222fcfc
0222fca4  7d61f1f8 ntdll!_except_handler3
0222fca8  7d628148 ntdll!`string'+0xac
0222fcac  ffffffff
0222fcb0  7d62821c ntdll!LdrpInitializeThread+0x18f
```

```
0222fcb4  7d61e299 ntdll!ZwTestAlert+0x15
0222fcb8  7d628088 ntdll!_LdrpInitialize+0x1de
0222fcbc  0222fd20
0222fcc0  00000000
...
0222fcfc  0222ffec
0222fd00  7d61f1f8 ntdll!_except_handler3
0222fd04  7d628090 ntdll!`string'+0xfc
0222fd08  ffffffff
0222fd0c  7d628088 ntdll!_LdrpInitialize+0x1de
0222fd10  7d61ce0d ntdll!NtContinue+0x12
0222fd14  7d61e9b2 ntdll!KiUserApcDispatcher+0x3a
0222fd18  0222fd20
0222fd1c  00000001
0222fd20  0001002f
...
0222fdc8  00000000
0222fdcc  00000000
0222fdd0  00411032 NullThread!ILT+45(?ThreadProcYGKPAXZ)
0222fdd4  00000000
0222fdd8  7d4d1504 kernel32!BaseThreadStartThunk
0222fddc  00000023
0222fde0  00000202
...
0222ffb4  cccccccc
0222ffb8  0222ffec
0222ffbc  7d4dfe21 kernel32!BaseThreadStart+0x34
0222ffc0  00000000
0222ffc4  00000000
0222ffc8  00000000
0222ffcc  00000000
0222ffd0  00000000
0222ffd4  0222ffc4
0222ffd8  00000000
0222ffdc  ffffffff
0222ffe0  7d4d89c4 kernel32!_except_handler3
0222ffe4  7d4dfe28 kernel32!`string'+0x18
0222ffe8  00000000
0222ffec  00000000
0222fff0  00000000
0222fff4  00411032 NullThread!ILT+45(?ThreadProcYGKPAXZ)
0222fff8  00000000
0222fffc  00000000
02230000  ????????
```

The second crashed thread has much more symbolic information in it, overwriting previous thread startup residue. It is mostly the exception handling residue because exception handling consumes stack space, as explained in the article **Who calls the postmortem debugger?** (Volume 1, page 113):

```
0:003> dds 0236a000 02370000
0236a000  00000000
...
0236a060  00000000
0236a064  0236a074
0236a068  00220000
0236a06c  7d61f7b4 ntdll!RtlpAllocateFromHeapLookaside+0x13
0236a070  00221378
0236a074  0236a29c
0236a078  7d61f748 ntdll!RtlAllocateHeap+0x1dd
0236a07c  7d61f78c ntdll!RtlAllocateHeap+0xee7
0236a080  0236a5f4
```

```
0236a084  00000000
...
0236a1b4  0236a300
0236a1b8  0236a1dc
0236a1bc  7d624267 ntdll!RtlIsDosDeviceName_Ustr+0x2f
0236a1c0  0236a21c
0236a1c4  7d624274 ntdll!RtlpDosSlashCONDevice
0236a1c8  00000001
0236a1cc  0236a317
0236a1d0  00000000
0236a1d4  0236a324
0236a1d8  0236a290
0236a1dc  7d6248af ntdll!RtlGetFullPathName_Ustr+0x80b
0236a1e0  7d6a00e0 ntdll!FastPebLock
0236a1e4  7d62489d ntdll!RtlGetFullPathName_Ustr+0x15b
0236a1e8  0236a5f4
0236a1ec  00000208
...
0236a224  00000000
0236a228  00000038
0236a22c  02080038 oleaut32!_PictSaveMetaFile+0x33
0236a230  00000000
...
0236a27c  00000000
0236a280  0236a53c
0236a284  7d61f1f8 ntdll!_except_handler3
0236a288  7d6245f0 ntdll!`string'+0x5c
0236a28c  ffffffff
0236a290  7d62489d ntdll!RtlGetFullPathName_Ustr+0x15b
0236a294  0236a5c8
0236a298  00000008
0236a29c  00000000
0236a2a0  0236a54c
0236a2a4  7d624bcf ntdll!RtlpDosPathNameToRelativeNtPathName_Ustr+0x3d8
0236a2a8  7d6a00e0 ntdll!FastPebLock
0236a2ac  7d624ba1 ntdll!RtlpDosPathNameToRelativeNtPathName_Ustr+0x3cb
0236a2b0  00000000
0236a2b4  0236e6d0
...
0236a2e0  000a0008
0236a2e4  7d624be8 ntdll!`string'
0236a2e8  00000000
0236a2ec  003a0038
...
0236a330  00650070
0236a334  0050005c
0236a338  00480043 advapi32!LsaGetQuotasForAccount+0x25
0236a33c  00610046
0236a340  006c0075
0236a344  00520074
0236a348  00700065
0236a34c  00780045
0236a350  00630065
0236a354  00690050
0236a358  00650070
0236a35c  00000000
0236a360  00000000
...
0236a4a0  0236a4b0
0236a4a4  00000001
0236a4a8  7d61f645 ntdll!RtlpFreeToHeapLookaside+0x22
0236a4ac  00230b98
```

```
0236a4b0  0236a590
0236a4b4  7d61f5d1 ntdll!RtlFreeHeap+0x20e
0236a4b8  00221378
0236a4bc  7d61f5ed ntdll!RtlFreeHeap+0x70f
0236a4c0  00000000
0236a4c4  7d61f4ab ntdll!RtlFreeHeap
0236a4c8  00000000
0236a4cc  00000000
...
0236a538  00000000
0236a53c  0236a678
0236a540  7d61f1f8 ntdll!_except_handler3
0236a544  7d624ba8 ntdll!`string'+0x1c
0236a548  ffffffff
0236a54c  7d624ba1 ntdll!RtlpDosPathNameToRelativeNtPathName_Ustr+0x3cb
0236a550  7d624c43 ntdll!RtlpDosPathNameToRelativeNtPathName_U+0x55
0236a554  00000001
0236a558  0236a56c
...
0236a590  0236a5c0
0236a594  7d620304 ntdll!RtlNtStatusToDosError+0x38
0236a598  7d620309 ntdll!RtlNtStatusToDosError+0x3d
0236a59c  7d61c828 ntdll!ZwWaitForSingleObject+0x15
0236a5a0  7d4d8c82 kernel32!WaitForSingleObjectEx+0xac
0236a5a4  00000124
0236a5a8  00000000
0236a5ac  7d4d8ca7 kernel32!WaitForSingleObjectEx+0xdc
0236a5b0  00000124
0236a5b4  7d61f49c ntdll!RtlGetLastWin32Error
0236a5b8  80070000
0236a5bc  00000024
...
0236a5f8  00000000
0236a5fc  0236a678
0236a600  7d4d89c4 kernel32!_except_handler3
0236a604  7d4d8cb0 kernel32!`string'+0x68
0236a608  ffffffff
0236a60c  7d4d8ca7 kernel32!WaitForSingleObjectEx+0xdc
0236a610  7d4d8bf1 kernel32!WaitForSingleObject+0x12
0236a614  7d61f49c ntdll!RtlGetLastWin32Error
0236a618  7d61c92d ntdll!NtClose+0x12
0236a61c  7d4d8e4f kernel32!CloseHandle+0x59
0236a620  00000124
0236a624  0236a688
0236a628  69511753 <Unloaded_faultrep.dll>+0x11753
0236a62c  6951175b <Unloaded_faultrep.dll>+0x1175b
0236a630  0236c6d0
...
0236a668  00000120
0236a66c  00000000
0236a670  0236a630
0236a674  7d94a2e9 user32!GetSystemMetrics+0x62
0236a678  0236f920
0236a67c  69510078 <Unloaded_faultrep.dll>+0x10078
0236a680  69503d10 <Unloaded_faultrep.dll>+0x3d10
0236a684  ffffffff
0236a688  6951175b <Unloaded_faultrep.dll>+0x1175b
0236a68c  69506136 <Unloaded_faultrep.dll>+0x6136
0236a690  0236e6d0
0236a694  0236c6d0
0236a698  0000009c
0236a69c  0236a6d0
```

```
0236a6a0  00002000
0236a6a4  0236eae4
0236a6a8  695061ff <Unloaded_faultrep.dll>+0x61ff
0236a6ac  00000000
0236a6b0  00000001
0236a6b4  0236f742
0236a6b8  69506210 <Unloaded_faultrep.dll>+0x6210
0236a6bc  00000028
0236a6c0  0236c76c
...
0236e6e0  0050005c
0236e6e4  00480043 advapi32!LsaGetQuotasForAccount+0x25
0236e6e8  00610046
...
0236e718  002204d8
0236e71c  0236e890
0236e720  77b940bb <Unloaded_VERSION.dll>+0x40bb
0236e724  77b91798 <Unloaded_VERSION.dll>+0x1798
0236e728  ffffffff
0236e72c  77b9178e <Unloaded_VERSION.dll>+0x178e
0236e730  69512587 <Unloaded_faultrep.dll>+0x12587
0236e734  0236e744
0236e738  00220000
0236e73c  7d61f7b4 ntdll!RtlpAllocateFromHeapLookaside+0x13
0236e740  00221378
0236e744  0236e96c
0236e748  7d61f748 ntdll!RtlAllocateHeap+0x1dd
0236e74c  7d61f78c ntdll!RtlAllocateHeap+0xee7
0236e750  0236eca4
0236e754  00000000
0236e758  0236ec94
0236e75c  7d620309 ntdll!RtlNtStatusToDosError+0x3d
0236e760  0236e7c8
0236e764  7d61c9db ntdll!NtQueryValueKey
0236e768  0236e888
0236e76c  0236e760
0236e770  7d61c9ed ntdll!NtQueryValueKey+0x12
0236e774  0236f920
0236e778  7d61f1f8 ntdll!_except_handler3
0236e77c  7d620310 ntdll!RtlpRunTable+0x490
0236e780  0236e790
0236e784  00220000
0236e788  7d61f7b4 ntdll!RtlpAllocateFromHeapLookaside+0x13
0236e78c  00221378
0236e790  0236e9b8
0236e794  7d61f748 ntdll!RtlAllocateHeap+0x1dd
0236e798  7d61f78c ntdll!RtlAllocateHeap+0xee7
0236e79c  0236ef18
0236e7a0  00000000
0236e7a4  00000000
0236e7a8  00220000
0236e7ac  0236e89c
0236e7b0  00000000
0236e7b4  00000128
0236e7b8  00000000
0236e7bc  0236e8c8
0236e7c0  0236e7c8
0236e7c4  c0000034
0236e7c8  0236e814
0236e7cc  7d61f1f8 ntdll!_except_handler3
0236e7d0  7d61f5f0 ntdll!CheckHeapFillPattern+0x64
0236e7d4  ffffffff
```

```
0236e7d8  7d61f5ed ntdll!RtlFreeHeap+0x70f
0236e7dc  7d4ded95 kernel32!FindClose+0x9b
0236e7e0  00220000
0236e7e4  00000000
0236e7e8  00220000
0236e7ec  00000000
0236e7f0  002314b4
0236e7f4  7d61ca1d ntdll!NtQueryInformationProcess+0x12
0236e7f8  7d4da465 kernel32!GetErrorMode+0x18
0236e7fc  ffffffff
0236e800  0000000c
0236e804  7d61ca65 ntdll!ZwSetInformationProcess+0x12
0236e808  7d4da441 kernel32!SetErrorMode+0x37
0236e80c  ffffffff
0236e810  0000000c
0236e814  0236e820
0236e818  00000004
0236e81c  00000000
0236e820  00000005
0236e824  0236eae8
0236e828  7d4e445f kernel32!GetLongPathNameW+0x38f
0236e82c  7d4e4472 kernel32!GetLongPathNameW+0x3a2
0236e830  00000001
0236e834  00000103
0236e838  00000000
0236e83c  0236f712
0236e840  7efaf000
0236e844  002316f0
0236e848  0000005c
0236e84c  7efaf000
0236e850  00000004
0236e854  002314b4
0236e858  0000ea13
0236e85c  0236e894
0236e860  00456b0d advapi32!RegQueryValueExW+0x96
0236e864  00000128
0236e868  0236e888
0236e86c  0236e8ac
0236e870  0236e8c8
0236e874  0236e8a4
0236e878  0236e89c
0236e87c  0236e88c
0236e880  7d635dc4 ntdll!iswdigit+0xf
0236e884  00000064
0236e888  00000004
0236e88c  7d624d81 ntdll!RtlpValidateCurrentDirectory+0xf6
0236e890  7d635d4e ntdll!RtlIsDosDeviceName_Ustr+0x1c0
0236e894  00000064
0236e898  0236e9d0
0236e89c  0236e9e7
0236e8a0  00000000
0236e8a4  0236e9f4
0236e8a8  0236e960
0236e8ac  7d6248af ntdll!RtlGetFullPathName_Ustr+0x80b
0236e8b0  7d6a00e0 ntdll!FastPebLock
0236e8b4  7d62489d ntdll!RtlGetFullPathName_Ustr+0x15b
0236e8b8  0236eca4
0236e8bc  00000208
0236e8c0  0236ec94
0236e8c4  00000000
0236e8c8  00220178
0236e8cc  00000004
```

```
0236e8d0  0236eb3c
0236e8d4  0236e8c8
0236e8d8  7d624d81 ntdll!RtlpValidateCurrentDirectory+0xf6
0236e8dc  0236e8f8
0236e8e0  7d6246c1 ntdll!RtlIsDosDeviceName_Ustr+0x14
0236e8e4  0236ea1c
0236e8e8  0236ea33
0236e8ec  00000000
0236e8f0  0236ea40
0236e8f4  0236e9ac
0236e8f8  7d6248af ntdll!RtlGetFullPathName_Ustr+0x80b
0236e8fc  7d6a00e0 ntdll!FastPebLock
0236e900  7d62489d ntdll!RtlGetFullPathName_Ustr+0x15b
0236e904  0236ef18
0236e908  00000208
...
0236e934  00000022
0236e938  00460044 advapi32!GetPerflibKeyValue+0x19e
0236e93c  0236ecd0
0236e940  00000000
0236e944  00000044
0236e948  02080044 oleaut32!_PictSaveMetaFile+0x3f
0236e94c  00000000
0236e950  4336ec0c
...
0236e9a8  0236ebd0
0236e9ac  7d62155b ntdll!RtlAllocateHeap+0x460
0236e9b0  7d61f78c ntdll!RtlAllocateHeap+0xee7
0236e9b4  00000000
0236e9b8  000003ee
0236e9bc  0236ed2c
0236e9c0  7d624bcf ntdll!RtlpDosPathNameToRelativeNtPathName_Ustr+0x3d8
0236e9c4  7d6a00e0 ntdll!FastPebLock
0236e9c8  00000ab0
0236e9cc  00000381
0236e9d0  00233950
0236e9d4  0236ebfc
0236e9d8  7d62155b ntdll!RtlAllocateHeap+0x460
0236e9dc  7d61f78c ntdll!RtlAllocateHeap+0xee7
0236e9e0  00000003
0236e9e4  fffffffc
0236e9e8  00000aa4
0236e9ec  00230ba0
0236e9f0  00000004
0236e9f4  003a0043
0236e9f8  00000000
0236e9fc  000a0008
0236ea00  7d624be8 ntdll!`string'
0236ea04  00000000
0236ea08  00460044 advapi32!GetPerflibKeyValue+0x19e
0236ea0c  0236ecd0
0236ea10  00233948
...
0236ea44  00220640
0236ea48  7d62273d ntdll!RtlIntegerToUnicode+0x126
0236ea4c  0000000c
...
0236eab4  0236f79c
0236eab8  7d61f1f8 ntdll!_except_handler3
0236eabc  7d622758 ntdll!RtlpIntegerWChars+0x54
0236eac0  00220178
0236eac4  0236ed3c
```

```
0236eac8  00000005
0236eacc  0236ed00
0236ead0  7d622660 ntdll!RtlConvertSidToUnicodeString+0x1cb
0236ead4  00220178
0236ead8  0236eaf0
0236eadc  0236eaec
0236eae0  00000001
0236eae4  7d61f645 ntdll!RtlpFreeToHeapLookaside+0x22
0236eae8  00223620
0236eaec  00220178
0236eaf0  7d61f5d1 ntdll!RtlFreeHeap+0x20e
0236eaf4  002217f8
0236eaf8  7d61f5ed ntdll!RtlFreeHeap+0x70f
0236eafc  00000000
0236eb00  00220178
...
0236eb48  0236eb58
0236eb4c  7d635dc4 ntdll!iswdigit+0xf
0236eb50  00220178
0236eb54  00000381
0236eb58  002343f8
0236eb5c  0236eb78
0236eb60  7d620deb ntdll!RtlpCoalesceFreeBlocks+0x383
0236eb64  00000381
0236eb68  002343f8
0236eb6c  00220000
0236eb70  00233948
0236eb74  00220000
0236eb78  00000000
0236eb7c  00220000
0236eb80  0236ec60
0236eb84  7d620fbe ntdll!RtlFreeHeap+0x6b0
0236eb88  00220608
0236eb8c  7d61f5ed ntdll!RtlFreeHeap+0x70f
0236eb90  000000e8
0236eb94  7d61cd23 ntdll!ZwWriteVirtualMemory
0236eb98  7efde000
0236eb9c  000000e8
0236eba0  00233948
0236eba4  7efde000
0236eba8  000002e8
0236ebac  0000005d
0236ebb0  00220178
0236ebb4  00000156
0236ebb8  0236e9b4
0236ebbc  00233948
0236ebc0  7d61f1f8 ntdll!_except_handler3
0236ebc4  00000ab0
0236ebc8  00233948
0236ebcc  00233950
0236ebd0  00220178
0236ebd4  00220000
0236ebd8  00000ab0
0236ebdc  00220178
0236ebe0  00000000
0236ebe4  00233950
0236ebe8  7d4ddea8 kernel32!`string'+0x50
0236ebec  00000000
0236ebf0  00233950
0236ebf4  00220178
0236ebf8  00000aa4
0236ebfc  00000000
```

```
0236ec00  0236ec54
0236ec04  7d63668a ntdll!RtlCreateProcessParameters+0x375
0236ec08  7d63668f ntdll!RtlCreateProcessParameters+0x37a
0236ec0c  7d6369e9 ntdll!RtlCreateProcessParameters+0x35f
0236ec10  00000000
...
0236ec4c  0000007f
0236ec50  0236ef4c
0236ec54  7d61f1f8 ntdll!_except_handler3
0236ec58  7d61f5f0 ntdll!CheckHeapFillPattern+0x64
0236ec5c  ffffffff
0236ec60  7d61f5ed ntdll!RtlFreeHeap+0x70f
0236ec64  7d6365e2 ntdll!RtlDestroyProcessParameters+0x1b
0236ec68  00220000
0236ec6c  00000000
0236ec70  00233950
0236ec74  0236ef5c
0236ec78  7d4ec4bc kernel32!BasePushProcessParameters+0x806
0236ec7c  00233950
0236ec80  7d4ec478 kernel32!BasePushProcessParameters+0x7c5
0236ec84  7efde000
0236ec88  0236f748
0236ec8c  00000000
0236ec90  0236ed92
0236ec94  00000000
0236ec98  00000000
0236ec9c  01060104
0236eca0  0236f814
0236eca4  0020001e
0236eca8  7d535b50 kernel32!`string'
0236ecac  00780076
0236ecb0  002314e0
0236ecb4  00780076
0236ecb8  0236ed2c
0236ecbc  00020000
0236ecc0  7d4ddee4 kernel32!`string'
0236ecc4  0236efec
...
0236ed3c  006d0061
0236ed40  00460020 advapi32!GetPerflibKeyValue+0x17a
0236ed44  006c0069
0236ed48  00730065
0236ed4c  00280020
0236ed50  00380078
0236ed54  00290036
0236ed58  0044005c advapi32!CryptDuplicateHash+0x3
0236ed5c  00620065
0236ed60  00670075
...
0236ee7c  0236ee8c
0236ee80  00000001
0236ee84  7d61f645 ntdll!RtlpFreeToHeapLookaside+0x22
0236ee88  00230dc0
0236ee8c  0236ef6c
0236ee90  0236eea0
0236ee94  00000001
0236ee98  7d61f645 ntdll!RtlpFreeToHeapLookaside+0x22
0236ee9c  00223908
0236eea0  0236ef80
0236eea4  7d61f5d1 ntdll!RtlFreeHeap+0x20e
0236eea8  00221d38
0236eeac  7d61f5ed ntdll!RtlFreeHeap+0x70f
```

```
0236eeb0  7d61f4ab ntdll!RtlFreeHeap
0236eeb4  7d61c91b ntdll!NtClose
0236eeb8  00000000
...
0236ef08  00000000
0236ef0c  7d621954 ntdll!RtlImageNtHeaderEx+0xee
0236ef10  7efde000
0236ef14  00001000
0236ef18  00000000
0236ef1c  000000e8
0236ef20  004000e8 NullThread!_enc$textbss$begin <PERF> (NullThread+0xe8)
0236ef24  00000000
0236ef28  0236ef10
0236ef2c  00000000
0236ef30  0236f79c
0236ef34  7d61f1f8 ntdll!_except_handler3
0236ef38  7d621954 ntdll!RtlImageNtHeaderEx+0xee
0236ef3c  00220000
...
0236ef68  0236eeb0
0236ef6c  7d61f5ed ntdll!RtlFreeHeap+0x70f
0236ef70  0236f79c
0236ef74  7d61f1f8 ntdll!_except_handler3
0236ef78  7d61f5f0 ntdll!CheckHeapFillPattern+0x64
0236ef7c  ffffffff
0236ef80  7d61f5ed ntdll!RtlFreeHeap+0x70f
0236ef84  7d4ea183 kernel32!CreateProcessInternalW+0x21f5
0236ef88  00220000
0236ef8c  00000000
0236ef90  00223910
0236ef94  7d4ebc0b kernel32!CreateProcessInternalW+0x1f26
0236ef98  00000000
0236ef9c  00000096
0236efa0  0236f814
0236efa4  00000103
0236efa8  7efde000
0236efac  00000001
0236efb0  0236effc
0236efb4  00000200
0236efb8  00000cb0
0236efbc  0236f00c
0236efc0  0236efdc
0236efc4  7d6256e8 ntdll!bsearch+0x42
0236efc8  00180144
0236efcc  0236efe0
0236efd0  7d625992 ntdll!ARRAY_FITS+0x29
0236efd4  00000a8c
0236efd8  00000000
0236efdc  00000000
0236efe0  00080000
0236efe4  00070000
0236efe8  00040000
0236efec  00000044
0236eff0  00000000
0236eff4  7d535b50 kernel32!`string'
0236eff8  00000000
0236effc  00000000
...
0236f070  00000001
0236f074  7d625ad8 ntdll!RtlFindActivationContextSectionString+0xe1
0236f078  004000e8 NullThread!_enc$textbss$begin <PERF> (NullThread+0xe8)
0236f07c  0236f0cc
```

```
0236f080  00000000
0236f084  7d6256e8 ntdll!bsearch+0x42
0236f088  00180144
0236f08c  0236f0a0
0236f090  7d625992 ntdll!ARRAY_FITS+0x29
0236f094  00000a8c
...
0236f0d0  0236f120
0236f0d4  7d625b62 ntdll!RtlpFindUnicodeStringInSection+0x7b
0236f0d8  0236f204
0236f0dc  00000020
...
0236f190  000002a8
0236f194  7d625b62 ntdll!RtlpFindUnicodeStringInSection+0x7b
0236f198  00000001
0236f19c  00000000
0236f1a0  0236f1d0
0236f1a4  7d6257f1 ntdll!RtlpFindNextActivationContextSection+0x64
0236f1a8  00181f1c
...
0236f1f0  7efaf000
0236f1f4  7d625ad8 ntdll!RtlFindActivationContextSectionString+0xe1
0236f1f8  0236f214
0236f1fc  0236f24c
0236f200  00000000
0236f204  7d6256e8 ntdll!bsearch+0x42
0236f208  00180144
...
0236f24c  00000200
0236f250  00000734
0236f254  7d625b62 ntdll!RtlpFindUnicodeStringInSection+0x7b
0236f258  0236f384
...
0236f3f0  00000000
0236f3f4  00000000
0236f3f8  01034236
0236f3fc  00000000
0236f400  7d4d1510 kernel32!BaseProcessStartThunk
0236f404  00000018
0236f408  00003000
...
0236f62c  0236f63c
0236f630  00000001
0236f634  7d61f645 ntdll!RtlpFreeToHeapLookaside+0x22
0236f638  00231088
0236f63c  0236f71c
...
0236f70c  002333b8
0236f710  0236f720
0236f714  00000001
0236f718  7d61f645 ntdll!RtlpFreeToHeapLookaside+0x22
0236f71c  00228fb0
0236f720  0236f800
0236f724  7d61f5d1 ntdll!RtlFreeHeap+0x20e
0236f728  00221318
0236f72c  7d61f5ed ntdll!RtlFreeHeap+0x70f
0236f730  00000000
0236f734  00000096
0236f738  0236f814
0236f73c  00220608
0236f740  7d61f5ed ntdll!RtlFreeHeap+0x70f
0236f744  0236f904
```

```
0236f748  008e0000
0236f74c  002334c2
...
0236f784  0236f7bc
0236f788  7d63d275 ntdll!_vsnwprintf+0x30
0236f78c  0236f79c
0236f790  0000f949
0236f794  0236ef98
0236f798  00000095
0236f79c  0236fb7c
0236f7a0  7d4d89c4 kernel32!_except_handler3
0236f7a4  7d4ed1d0 kernel32!`string'+0xc
0236f7a8  ffffffff
0236f7ac  7d4ebc0b kernel32!CreateProcessInternalW+0x1f26
0236f7b0  7d4d14a2 kernel32!CreateProcessW+0x2c
0236f7b4  00000000
...
0236f7f0  0236fb7c
0236f7f4  7d61f1f8 ntdll!_except_handler3
0236f7f8  7d61d051 ntdll!NtWaitForMultipleObjects+0x15
0236f7fc  7d61c92d ntdll!NtClose+0x12
0236f800  7d4d8e4f kernel32!CloseHandle+0x59
0236f804  00000108
0236f808  0236fb8c
0236f80c  7d535b07 kernel32!UnhandledExceptionFilter+0x815
0236f810  00000108
0236f814  00430022 advapi32!_imp__OutputDebugStringW <PERF> (advapi32+0x22)
0236f818  005c003a
0236f81c  00720050
...
0236f8ec  0055005c
0236f8f0  00650073
0236f8f4  00440072 advapi32!CryptDuplicateHash+0x19
0236f8f8  006d0075
0236f8fc  00730070
0236f900  006e005c
0236f904  00770065
0236f908  0064002e
0236f90c  0070006d
0236f910  0020003b
0236f914  00220071
0236f918  00000000
0236f91c  00000096
0236f920  7d4dda47 kernel32!DuplicateHandle+0xd0
0236f924  7d4dda47 kernel32!DuplicateHandle+0xd0
0236f928  0236fb8c
0236f92c  7d5358cb kernel32!UnhandledExceptionFilter+0x5f1
0236f930  0236f9f0
0236f934  00000001
0236f938  00000000
0236f93c  7d535b43 kernel32!UnhandledExceptionFilter+0x851
0236f940  00000000
0236f944  00000000
0236f948  00000000
0236f94c  0236f95c
0236f950  00000098
0236f954  000001a2
0236f958  01c423b0
0236f95c  0236fb84
0236f960  7d62155b ntdll!RtlAllocateHeap+0x460
0236f964  7d61f78c ntdll!RtlAllocateHeap+0xee7
0236f968  00000000
```

```
0236f96c  0000008c
0236f970  00000000
0236f974  7d4d8472 kernel32!$$VProc_ImageExportDirectory+0x6d4e
0236f978  0236fa1c
0236f97c  00000044
0236f980  00000000
0236f984  7d535b50 kernel32!`string'
0236f988  00000000
0236f98c  00000000
0236f990  00000000
0236f994  00000000
0236f998  00000000
0236f99c  00000000
0236f9a0  00000000
0236f9a4  00000000
0236f9a8  00000000
0236f9ac  00000000
0236f9b0  00000000
0236f9b4  00000000
0236f9b8  00000000
0236f9bc  00000000
0236f9c0  0010000e
0236f9c4  7ffe0030 SharedUserData+0x30
0236f9c8  000000e8
0236f9cc  00000108
0236f9d0  00000200
0236f9d4  00000734
0236f9d8  00000018
0236f9dc  00000000
0236f9e0  7d5621d0 kernel32!ProgramFilesEnvironment+0x74
0236f9e4  00000040
0236f9e8  00000000
0236f9ec  00000000
0236f9f0  0000000c
0236f9f4  00000000
0236f9f8  00000001
0236f9fc  00000118
0236fa00  000000e8
0236fa04  c0000005
0236fa08  00000000
0236fa0c  00000008
0236fa10  00000000
0236fa14  00000110
0236fa18  0236f814
0236fa1c  6950878a <Unloaded_faultrep.dll>+0x878a
0236fa20  00120010
0236fa24  7d51c5e4 kernel32!`string'
0236fa28  00000003
0236fa2c  05bc0047
...
0236fa74  0057005c
0236fa78  004b0032 advapi32!szPerflibSectionName <PERF> (advapi32+0x80032)
0236fa7c  005c0033
0236fa80  00790073
...
0236fac8  0000002b
0236facc  00000000
0236fad0  7d61e3e6 ntdll!ZwWow64CsrNewThread+0x12
0236fad4  00000000
...
0236fb44  00000000
0236fb48  00000000
```

```
0236fb4c  7d61cb0d ntdll!ZwQueryVirtualMemory+0x12
0236fb50  7d54eeb8 kernel32!_ValidateEH3RN+0xb6
0236fb54  ffffffff
0236fb58  7d4dfe28 kernel32!`string'+0x18
0236fb5c  00000000
0236fb60  0236fb78
0236fb64  0000001c
0236fb68  0000000f
0236fb6c  7d4dfe28 kernel32!`string'+0x18
0236fb70  0000f949
0236fb74  0236f814
0236fb78  7d4df000 kernel32!CheckForSameCurdir+0x39
0236fb7c  0236fbd4
0236fb80  7d4d89c4 kernel32!_except_handler3
0236fb84  7d535be0 kernel32!`string'+0xc
0236fb88  ffffffff
0236fb8c  7d535b43 kernel32!UnhandledExceptionFilter+0x851
0236fb90  7d508f4e kernel32!BaseThreadStart+0x4a
0236fb94  0236fbb4
0236fb98  7d4d8a25 kernel32!_except_handler3+0x61
0236fb9c  0236fbbc
0236fba0  00000000
0236fba4  0236fbbc
0236fba8  00000000
0236fbac  00000000
0236fbb0  00000000
0236fbb4  0236fca0
0236fbb8  0236fcf0
0236fbbc  0236fbe0
0236fbc0  7d61ec2a ntdll!ExecuteHandler2+0x26
0236fbc4  0236fca0
0236fbc8  0236ffdc
0236fbcc  0236fcf0
0236fbd0  0236fc7c
0236fbd4  0236ffdc
0236fbd8  7d61ec3e ntdll!ExecuteHandler2+0x3a
0236fbdc  0236ffdc
0236fbe0  0236fc88
0236fbe4  7d61ebfb ntdll!ExecuteHandler+0x24
0236fbe8  0236fca0
0236fbec  0236ffdc
0236fbf0  00000000
0236fbf4  0236fc7c
0236fbf8  7d4d89c4 kernel32!_except_handler3
0236fbfc  00000000
0236fc00  0036fca0
0236fc04  0236fc18
0236fc08  7d640ca6 ntdll!RtlCallVectoredContinueHandlers+0x15
0236fc0c  0236fca0
0236fc10  0236fcf0
0236fc14  7d6a0608 ntdll!RtlpCallbackEntryList
0236fc18  0236fc88
0236fc1c  7d6354c9 ntdll!RtlDispatchException+0x11f
0236fc20  0236fca0
0236fc24  0236fcf0
0236fc28  00000000
0236fc2c  00000000
...
0236fc88  0236ffec
0236fc8c  7d61dd26 ntdll!NtRaiseException+0x12
0236fc90  7d61ea51 ntdll!KiUserExceptionDispatcher+0x29
0236fc94  0236fca0
```

```
0236fc98  0236fcf0
0236fc9c  00000000
0236fca0  c0000005
0236fca4  00000000
0236fca8  00000000
0236fcac  00000000
0236fcb0  00000002
0236fcb4  00000008
0236fcb8  00000000
0236fcbc  00000000
0236fcc0  00000000
0236fcc4  6b021fa0
0236fcc8  78b83980
0236fccc  00000000
0236fcd0  00000000
0236fcd4  00000000
0236fcd8  7efad000
0236fcdc  023afd00
0236fce0  023af110
0236fce4  78b83980
0236fce8  010402e1
0236fcec  00000000
0236fcf0  0001003f
0236fcf4  00000000
0236fcf8  00000000
0236fcfc  00000000
0236fd00  00000000
0236fd04  00000000
0236fd08  00000000
0236fd0c  0000027f
0236fd10  00000000
0236fd14  0000ffff
0236fd18  00000000
0236fd1c  00000000
0236fd20  00000000
0236fd24  00000000
0236fd28  00000000
0236fd2c  00000000
0236fd30  00000000
0236fd34  00000000
0236fd38  00000000
0236fd3c  00000000
0236fd40  00000000
0236fd44  00000000
0236fd48  00000000
0236fd4c  00000000
0236fd50  00000000
0236fd54  00000000
0236fd58  00000000
0236fd5c  00000000
0236fd60  00000000
0236fd64  00000000
0236fd68  00000000
0236fd6c  00000000
0236fd70  00000000
0236fd74  00000000
0236fd78  00000000
0236fd7c  0000002b
0236fd80  00000053
0236fd84  0000002b
0236fd88  0000002b
0236fd8c  00000000
```

```
0236fd90  00000000
0236fd94  00000000
0236fd98  00000000
0236fd9c  47f30000
0236fda0  00000000
0236fda4  0236ffec
0236fda8  00000000
0236fdac  00000023
0236fdb0  00010246
0236fdb4  0236ffbc
0236fdb8  0000002b
0236fdbc  0000027f
0236fdc0  00000000
0236fdc4  00000000
0236fdc8  00000000
0236fdcc  00000000
0236fdd0  00000000
0236fdd4  00001f80
0236fdd8  00000000
0236fddc  00000000
...
0236ffb4  00000000
0236ffb8  00000000
0236ffbc  7d4dfe21 kernel32!BaseThreadStart+0x34
0236ffc0  00000000
0236ffc4  00000000
0236ffc8  00000000
0236ffcc  00000000
0236ffd0  c0000005
0236ffd4  0236ffc4
0236ffd8  0236fbb4
0236ffdc  ffffffff
0236ffe0  7d4d89c4 kernel32!_except_handler3
0236ffe4  7d4dfe28 kernel32!`string'+0x18
0236ffe8  00000000
0236ffec  00000000
0236fff0  00000000
0236fff4  00000000
0236fff8  00000000
0236fffc  00000000
02370000  ????????
```

Fake Module

We started cataloging elemental malware detection and analysis patterns. The first such pattern is called **Deviant Module**. In **Fake Module** pattern, one of the loaded modules masquerades as a legitimate system DLL or a widely known value-adding DLL from some popular 3rd-party product. To illustrate this pattern, we modeled it as Victimware: a process crashed after loading a malware module:

```
0:000> k
*** Stack trace for last set context - .thread/.cxr resets it
Child-SP          RetAddr           Call Site
00000000`0026f978 00000001`3f89103a 0x0
00000000`0026f980 00000001`3f8911c4 FakeModule!wmain+0x3a
00000000`0026f9c0 00000000`76e3652d FakeModule!__tmainCRTStartup+0x144
00000000`0026fa00 00000000`7752c521 kernel32!BaseThreadInitThunk+0xd
00000000`0026fa30 00000000`00000000 ntdll!RtlUserThreadStart+0x1d
```

When we inspected loaded modules, we didn’t find anything suspicious:

```
0:000> lmp
start             end                 module name
00000000`76e20000 00000000`76f3f000   kernel32 <none>
00000000`77500000 00000000`776a9000   ntdll    <none>
00000001`3f890000 00000001`3f8a6000   FakeModule <none>
000007fe`f8cb0000 000007fe`f8cc7000   winspool <none>
000007fe`fdb30000 000007fe`fdb9c000   KERNELBASE <none>
```

However, when checking module images for any modifications we find that *winspool* module was not compared with the corresponding existing binary from Microsoft symbol server:

```
0:000> !for_each_module "!chkimg -v -d @#ModuleName"
Searching for module with expression: kernel32
Will apply relocation fixups to file used for comparison
Will ignore NOP/LOCK errors
Will ignore patched instructions
Image specific ignores will be applied
Comparison image path:
C:\WSDK8\Debuggers\x64\sym\kernel32.dll\503285C111f000\kernel32.dll
No range specified

Scanning section:    .text
Size: 633485
Range to scan: 76e21000-76ebba8d
Total bytes compared: 633485(100%)
Number of errors: 0
0 errors : kernel32
Searching for module with expression: ntdll
Will apply relocation fixups to file used for comparison
Will ignore NOP/LOCK errors
Will ignore patched instructions
Image specific ignores will be applied
Comparison image path: C:\WSDK8\Debuggers\x64\sym\ntdll.dll\4EC4AA8E1a9000\ntdll.dll
No range specified
```

```
Scanning section:    .text
Size: 1049210
Range to scan: 77501000-7760127a
Total bytes compared: 1049210(100%)
Number of errors: 0

Scanning section:       RT
Size: 474
Range to scan: 77602000-776021da
Total bytes compared: 474(100%)
Number of errors: 0
0 errors : ntdll
Searching for module with expression: FakeModule
Error for FakeModule: Could not find image file for the module. Make sure binaries are
included in the symbol path.
Searching for module with expression: winspool
Error for winspool: Could not find image file for the module. Make sure binaries are
included in the symbol path.
Searching for module with expression: KERNELBASE
Will apply relocation fixups to file used for comparison
Will ignore NOP/LOCK errors
Will ignore patched instructions
Image specific ignores will be applied
Comparison image path:
C:\WSDK8\Debuggers\x64\sym\KERNELBASE.dll\503285C26c000\KERNELBASE.dll
No range specified

Scanning section:    .text
Size: 302047
Range to scan: 7fefdb31000-7fefdb7abdf
Total bytes compared: 302047(100%)
Number of errors: 0
0 errors : KERNELBASE
```

Checking module data reveals that it was loaded not from the *System32* folder and also doesn't have any version information:

```
0:000> lmv m winspool
start             end                 module name
000007fe`f8cb0000 000007fe`f8cc7000   winspool   (deferred)
Image path: C:\Work\AWMA\FakeModule\x64\Release\winspool.drv
Image name: winspool.drv
Timestamp:        Fri Dec 28 22:22:42 2012 (50DE1BB2)
CheckSum:         00000000
ImageSize:        00017000
File version:     0.0.0.0
Product version:  0.0.0.0
File flags:       0 (Mask 0)
File OS:          0 Unknown Base
File type:        0.0 Unknown
File date:        00000000.00000000
Translations:     0000.04b0 0000.04e4 0409.04b0 0409.04e4
```

We could see that path from running the following command as well:

```
0:000> !for_each_module
00: 0000000076e20000  0000000076f3f000         kernel32
C:\Windows\System32\kernel32.dll                      kernel32.dll
01: 0000000077500000  00000000776a9000            ntdll
C:\Windows\System32\ntdll.dll                            ntdll.dll
02: 000000013f890000  000000013f8a6000       FakeModule
C:\Work\AWMA\FakeModule\x64\Release\FakeModule.exe  FakeModule.exe
03: 000007fef8cb0000  000007fef8cc7000         winspool
C:\Work\AWMA\FakeModule\x64\Release\winspool.drv
04: 000007fefdb30000  000007fefdb9c000       KERNELBASE
C:\Windows\System32\KERNELBASE.dll                  KERNELBASE.dll
```

Or from PEB:

```
0:000> !peb
PEB at 000007fffffdf000
[...]
7fef8cb0000 50de1bb2 Dec 28 22:22:42 2012
C:\Work\AWMA\FakeModule\x64\Release\winspool.drv
[...]
```

Another sign is the module size in memory which is much smaller than the real *winspool.drv*:

```
0:000> ? 000007fe`f8cc7000 - 000007fe`f8cb0000
Evaluate expression: 94208 = 00000000`0001700
```

Module size can help if a legitimate module from the well-known folder was replaced. Module debug directory and the size of export and import directories are also different from the original one revealing the development folder:

```
0:000> !dh 000007fe`f8cb0000
[...]
   0 [       0] address [size] of Export Directory
[...]
9000 [     208] address [size] of Import Address Table Directory
[...]
Debug Directories(2)
Type       Size     Address  Pointer
cv           49        e2c0     cac0 Format: RSDS, guid, 1,
C:\Work\AWMA\FakeModule\x64\Release\winspool.pdb
```

This can also be seen from the output of **!lmi** command:

```
0:000> !lmi 7fef8cb0000
Loaded Module Info: [7fef8cb0000]
Module: winspool
Base Address: 000007fef8cb0000
Image Name: winspool.drv
Machine Type: 34404 (X64)
Time Stamp: 50de1bb2 Fri Dec 28 22:22:42 2012
Size: 17000
```

```
CheckSum: 0
Characteristics: 2022
Debug Data Dirs: Type  Size     VA  Pointer
CODEVIEW    49,  e2c0,    cac0 RSDS - GUID: {29D85193-1C9D-4997-95BA-DD190FA3C1BF}
Age: 1, Pdb: C:\Work\AWMA\FakeModule\x64\Release\winspool.pdb
??    10,  e30c,    cb0c [Data not mapped]
Symbol Type: DEFERRED - No error - symbol load deferred
Load Report: no symbols loaded
```

Hidden Module

Sometimes we look for modules that were loaded and unloaded at some time. The **lm** command lists unloaded modules, but some of them could be mapped to address space without using the runtime loader. The latter case is common for DRM-type protection tools, rootkits, malware, or crimeware which can influence a process execution. In such cases, we can hope they still remain in virtual memory and search for them. WinDbg **.imgscan** command greatly helps in identifying MZ/PE module headers. The following example illustrates this command without implying that the found module did any harm:

```
0:000> .imgscan
MZ at 000d0000, prot 00000002, type 01000000 - size 6000
  Name: usrxcptn.dll
MZ at 00350000, prot 00000002, type 01000000 - size 9b000
  Name: ADVAPI32.dll
MZ at 00400000, prot 00000002, type 01000000 - size 23000
  Name: javaw.exe
MZ at 01df0000, prot 00000002, type 01000000 - size 8b000
  Name: OLEAUT32.dll
MZ at 01e80000, prot 00000002, type 01000000 - size 52000
  Name: SHLWAPI.dll
...
```

We don't see **usrxcptn** in either loaded or unloaded module lists:

```
0:002> lm
start    end        module name
00350000 003eb000   advapi32
00400000 00423000   javaw
01df0000 01e7b000   oleaut32
01e80000 01ed2000   shlwapi
...

Unloaded modules:
```

Then we can use **Unknown Component** pattern (Volume 1, page 367) to see the module resources if present in memory:

```
0:002> !dh 000d0000

...

SECTION HEADER #4
   .rsrc name
     418 virtual size
    4000 virtual address
     600 size of raw data
    1600 file pointer to raw data
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
40000040 flags
         Initialized Data
         (no align specified)
         Read Only

...
```

```
0:002> dc 000d0000+4000 L418
...
000d4140  ... n…z.)…F.i.l.
000d4150  ... e.D.e.s.c.r.i.p.
000d4160  ... t.i.o.n…..U.s.
000d4170  ...   e.r. .D.u.m.p. .
000d4180  ... U.s.e.r. .M.o.d.
000d4190  ... e. .E.x.c.e.p.t.
000d41a0  ... i.o.n. .D.i.s.p.
000d41b0  ... a.t.c.h.e.r…..

0:002> du 000d416C
000d416c  "User Dump User Mode Exception Di"
000d41ac  "spatcher"
```

This component seems to be loaded or mapped only if userdump package was fully installed where usrxcptn.dll is a part of its redistribution, and the application was added to Process Dumper applet in Control Panel. Although from the memory dump comment, we also see that the dump was taken manually using the command line userdump.exe we see that the full userdump package was additionally installed, which was probably not necessary (see **Correcting Microsoft Article About userdump.exe**, Volume 1, page 612):

```
Loading Dump File [javaw.dmp]
User Mini Dump File with Full Memory: Only application data is available

Comment: 'Userdump generated complete user-mode minidump with Standalone function on
COMPUTER-NAME'
```

Hidden Process

Not all processes are linked into a list that some commands traverse, such as **!process 0 0**. A process may unlink itself or be in an initialization stage. However, a process structure is allocated from the nonpaged pool, and such pool can be searched for ”*Proc*” pool tag (unless a process changes that in memory). For example:

```
0: kd> !poolfind Proc

Searching NonPaged pool (83c3c000 : 8bc00000) for Tag: Proc

*87b15000 size:  298 previous size:    0  (Free)       Pro.
*87b18370 size:  298 previous size:   98  (Allocated)  Proc (Protected)
[...]
*8a35e900 size:  298 previous size:   30  (Allocated)  Proc (Protected)
*8a484000 size:  298 previous size:    0  (Allocated)  Proc (Protected)
*8a4a2d68 size:  298 previous size:   28  (Allocated)  Proc (Protected)
[...]
```

One such structure is missing from the active process linked list (note that it has a parent PID):

```
0: kd> !process 8a484000+20
PROCESS 8a484020  SessionId: 0  Cid: 05a0    Peb: 00000000  ParentCid: 0244
DirBase: bffc2200  ObjectTable: e17e6a78  HandleCount:   0.
Image: AppChild.exe
VadRoot 8a574f80 Vads 4 Clone 0 Private 3. Modified 0. Locked 0.
DeviceMap e1002898
Token                             e1a36030
ElapsedTime                       00:00:00.000
UserTime                          00:00:00.000
KernelTime                        419 Days 13:24:16.625
QuotaPoolUsage[PagedPool]         7580
QuotaPoolUsage[NonPagedPool]      160
Working Set Sizes (now,min,max)  (12, 50, 345) (48KB, 200KB, 1380KB)
PeakWorkingSetSize                12
VirtualSize                       1 Mb
PeakVirtualSize                   1 Mb
PageFaultCount                    5
MemoryPriority                    BACKGROUND
BasePriority                      8
CommitCharge                      156

No active threads
```

We may think that this process is a zombie (note that, unlike terminated processes, it has non-zero data such as VAD and object table and zero PEB and elapsed time), but inspection of its parent process thread stacks reveals that it was in the process of creation (note an attached process field):

```
THREAD 8a35dad8  Cid 0244.0248  Teb: 7ffdd000 Win32Thread: bc3aa688 WAIT: (Unknown)
KernelMode Non-Alertable
ba971608  NotificationEvent
Impersonation token:  e2285030 (Level Impersonation)
DeviceMap                 e1a31a58
Owning Process            8a35e920       Image:         AppParent.exe
Attached Process          8a484020       Image:         AppChild.exe
Wait Start TickCount      2099           Ticks: 1 (0:00:00:00.015)
Context Switch Count      279                 LargeStack
UserTime                  00:00:00.046
KernelTime                00:00:00.046
Win32 Start Address AppParent!mainCRTStartup (0x0100d303)
Start Address kernel32!BaseProcessStartThunk (0x77e617f8)
Stack Init ba972000 Current ba971364 Base ba972000 Limit ba96e000 Call 0
Priority 8 BasePriority 8 PriorityDecrement 0
ChildEBP RetAddr
ba97137c 80833f2d nt!KiSwapContext+0x26
ba9713a8 80829c72 nt!KiSwapThread+0x2e5
ba9713f0 bad3c9db nt!KeWaitForSingleObject+0x346
[...]
ba971b94 8094cfc3 nt!MmCreatePeb+0x2cc
ba971ce4 8094d42d nt!PspCreateProcess+0x5a9
ba971d38 8088b4ac nt!NtCreateProcessEx+0x77
ba971d38 7c82845c nt!KiFastCallEntry+0xfc (TrapFrame @ ba971d64)
0006f498 7c826d09 ntdll!KiFastSystemCallRet
0006f49c 77e6cf95 ntdll!ZwCreateProcessEx+0xc
0006fcc0 7d1ec670 kernel32!CreateProcessInternalW+0x15e5
0006fd0c 01008bcf ADVAPI32!CreateProcessAsUserW+0x108
[...]
```

Hooksware

This word describes applications heavily dependent on various hooks that are either injected by normal Windows hooking mechanism, registry, or via more elaborate tricks like remote threads or code patching. There are various patterns in memory dumps that help in the detection, troubleshooting, and debugging of **hooksware**:

- **Hooked Functions** (Volume 1, page 468)

This is the primary detection mechanism for hooks that patch code.

- **Changed Environment** (Volume 1, page 283)

Loaded hooks shift other modules by changing their load address and, therefore, might expose dormant bugs.

- **Insufficient Memory** (Volume 2, page 210)

Hooks loaded in the middle of address space limit the maximum amount of memory that can be allocated at once. For example, various virtual machines, like Java, reserve a big chunk of memory at startup.

- **No Component Symbols** (Volume 1, page 298)

We can get an approximate picture of what a 3rd-party hook module does by looking at its import table or, in the case of patching, by looking at the list of deviations returned by the **.chkimg** command.

- **Unknown Component** (Volume 1, page 367)

This pattern might give an idea about the author of the hook.

- **Coincidental Symbolic Information** (Volume 1, page 390)

Sometimes hooks are loaded at round addresses like 0x10000000, and these values are very frequently used as flags or constants too.

- **Wild Code** (Volume 2, page 219)

When hooking goes wrong, the execution path goes into the wild territory.

- **Execution Residue** (Volume 3, page 239)

Here we can find various hooks that use normal Windows hooking mechanism. Sometimes, the search for "hook" words in the symbolic raw stack output of the **dds** command reveals them but beware of coincidental symbolic information. See also how to dump raw stack from process dump files (Volume 1, page 231) and complete memory dumps (Volume 1, page 236).

- **Hidden Module** (Volume 2, page 286)

Some hooks may hide themselves.

Namespace

As usual, a new pattern arises with the need to communicate analysis findings. Most often, when analyzing malware, we don’t have symbol files (**No Component Symbols**) for **Unknown Module**. By looking at IAT (if any present), we can guess the module purpose. Sometimes a module itself is not malicious but is used in a larger malicious context such as screen grabbing:

```
[...]
10002000  76376101 gdi32!CreateCompatibleDC
10002004  763793d6 gdi32!StretchBlt
10002008  76377461 gdi32!CreateDIBSection
1000200c  763762a0 gdi32!SelectObject
10002010  00000000
10002024  77429ced user32!ReleaseDC
10002028  77423ba7 user32!NtUserGetWindowDC
1000202c  77430e21 user32!GetWindowRect
10002030  00000000
10002034  744a75e9 GdiPlus!GdiplusStartup
10002038  744976dd GdiPlus!GdipSaveImageToStream
1000203c  744cdd38 GdiPlus!GdipGetImageEncodersSize
10002040  744971cf GdiPlus!GdipDisposeImage
10002044  744a8591 GdiPlus!GdipCreateBitmapFromHBITMAP
10002048  744cdbae GdiPlus!GdipGetImageEncoders
[...]
```

There are also cases where these API names are not in IAT but found as **String Hint** in raw data such LoadLibrary / GetProcAddress and even a group of modules themselves as a collective API:

```
[...]
00058e20  "kernel32.dll"
00058e3c  "user32.dll"
00058e54  "ws2_32.dll"
00058e6c  "ntdll.dll"
00058e80  "wininet.dll"
00058e98  "nspr4.dll"
00058eac  "ssl3.dll"
[...]
```

No Component Symbols

Another pattern that happens so often in crash dumps: **No Component Symbols**. In this case, we can guess what a component does by looking at its name, the overall thread stack where it is called, and also its import table. Here is an example. We have component.sys driver visible on some thread stack in a kernel dump, but we don’t know what that component can potentially do. Because we don’t have symbols, we cannot see its imported functions:

```
kd> x component!*
kd>
```

We use **!dh** command to dump its image headers:

```
kd> lmv m component
start             end                 module name
fffffadf`e0eb5000 fffffadf`e0ebc000   component   (no symbols)
    Loaded symbol image file: component.sys
    Image path: \??\C:\Component\x64\component.sys
    Image name: component.sys
    Timestamp:        Sat Jul 01 19:06:16 2006 (44A6B998)
    CheckSum:         000074EF
    ImageSize:        00007000
    Translations:     0000.04b0 0000.04e0 0409.04b0 0409.04e0

kd> !dh fffffadf`e0eb5000
File Type: EXECUTABLE IMAGE
FILE HEADER VALUES
    8664 machine (X64)
       6 number of sections
44A6B998 time date stamp Sat Jul 01 19:06:16 2006
       0 file pointer to symbol table
       0 number of symbols
      F0 size of optional header
      22 characteristics
            Executable
            App can handle >2gb addresses
OPTIONAL HEADER VALUES
     20B magic #
    8.00 linker version
     C00 size of code
     A00 size of initialized data
       0 size of uninitialized data
    5100 address of entry point
    1000 base of code
         ----- new -----
0000000000010000 image base
    1000 section alignment
     200 file alignment
       1 subsystem (Native)
    5.02 operating system version
    5.02 image version
    5.02 subsystem version
    7000 size of image
     400 size of headers
    74EF checksum
0000000000040000 size of stack reserve
0000000000001000 size of stack commit
0000000000100000 size of heap reserve
0000000000001000 size of heap commit
```

```
       0 [       0] address [size] of Export Directory
    51B0 [      28] address [size] of Import Directory
    6000 [     3B8] address [size] of Resource Directory
    4000 [      6C] address [size] of Exception Directory
       0 [       0] address [size] of Security Directory
       0 [       0] address [size] of Base Relocation Directory
    2090 [      1C] address [size] of Debug Directory
       0 [       0] address [size] of Description Directory
       0 [       0] address [size] of Special Directory
       0 [       0] address [size] of Thread Storage Directory
       0 [       0] address [size] of Load Configuration Directory
       0 [       0] address [size] of Bound Import Directory
    2000 [      88] address [size] of Import Address Table Directory
       0 [       0] address [size] of Delay Import Directory
       0 [       0] address [size] of COR20 Header Directory
       0 [       0] address [size] of Reserved Directory
...
...
...
```

Then we display the contents of the Import Address Table Directory using the **dps** command:

```
kd> dps fffffadf`e0eb5000+2000 fffffadf`e0eb5000+2000+88
fffffadf`e0eb7000  fffff800`01044370 nt!IoCompleteRequest
fffffadf`e0eb7008  fffff800`01019700 nt!IoDeleteDevice
fffffadf`e0eb7010  fffff800`012551a0 nt!IoDeleteSymbolicLink
fffffadf`e0eb7018  fffff800`01056a90 nt!MiResolveTransitionFault+0x7c2
fffffadf`e0eb7020  fffff800`0103a380 nt!ObDereferenceObject
fffffadf`e0eb7028  fffff800`0103ace0 nt!KeWaitForSingleObject
fffffadf`e0eb7030  fffff800`0103c570 nt!KeSetTimer
fffffadf`e0eb7038  fffff800`0102d070 nt!IoBuildPartialMdl+0x3
fffffadf`e0eb7040  fffff800`012d4480 nt!PsTerminateSystemThread
fffffadf`e0eb7048  fffff800`01041690 nt!KeBugCheckEx
fffffadf`e0eb7050  fffff800`010381b0 nt!KeInitializeTimer
fffffadf`e0eb7058  fffff800`0103ceb0 nt!ZwClose
fffffadf`e0eb7060  fffff800`012b39f0 nt!ObReferenceObjectByHandle
fffffadf`e0eb7068  fffff800`012b7380 nt!PsCreateSystemThread
fffffadf`e0eb7070  fffff800`01251f90 nt!FsRtlpIsDfsEnabled+0x114
fffffadf`e0eb7078  fffff800`01275160 nt!IoCreateDevice
fffffadf`e0eb7080  00000000`00000000
fffffadf`e0eb7088  00000000`00000000
```

We see that this driver under certain circumstances can bugcheck the system using *KeBugCheckEx*, it creates system thread(s) (*PsCreateSystemThread*) and uses timer(s) (*KeInitializeTimer*, *KeSetTimer*).

If we see *name+offset* in the import table (I think this is an effect of OMAP code optimization), we can get the function by using the **ln** command (list nearest symbols):

```
kd> ln fffff800`01056a90
(fffff800`01056760)   nt!MiResolveTransitionFault+0x7c2   |  (fffff800`01056a92)   nt!R
tlInitUnicodeString

kd> ln fffff800`01251f90
(fffff800`01251e90)   nt!FsRtlpIsDfsEnabled+0x114   |  (fffff800`01251f92)   nt!IoCreat
eSymbolicLink
```

This technique is useful if we have a bugcheck that happens when a driver calls certain functions or must call a certain function in pairs, like bugcheck 0x20:

```
kd> !analyze -show 0x20
KERNEL_APC_PENDING_DURING_EXIT (20)
The key data item is the thread's APC disable count. If this is non-zero, then this is
the source of the problem. The APC disable count is decremented each time a driver
calls KeEnterCriticalRegion, KeInitializeMutex, or FsRtlEnterFileSystem. The APC
disable count is incremented each time a driver calls KeLeaveCriticalRegion,
KeReleaseMutex, or FsRtlExitFileSystem. Since these calls should always be in pairs,
this value should be zero when a thread exits. A negative value indicates that a driver
has disabled APC calls without re-enabling them. A positive value indicates that the
reverse is true. If you ever see this error, be very suspicious of all drivers
installed on the machine — especially unusual or non-standard drivers. Third party file
system redirectors are especially suspicious since they do not generally receive the
heavy duty testing that NTFS, FAT, RDR, etc receive. This current IRQL should also be
0. If it is not, that a driver’s cancelation routine can cause this bugcheck by return-
ing at an elevated IRQL. Always attempt to note what you were doing/closing at the time
of the crash, and note all of the installed drivers at the time of the crash.  This
symptom is usually a severe bug in a third party driver.
```

Then we can see at least whether the suspicious driver could have potentially used those functions and if it imports one of them, we can see whether it imports the corresponding counterpart function.

No Component Symbols pattern can be easily identified in stack traces by huge function offsets or no exported functions at all:

```
STACK_TEXT:
WARNING: Stack unwind information not available. Following frames may be wrong.
00b2f42c 091607aa mydll!foo+0x8338
00b2f4cc 7c83ab9e mydll2+0x8fe3
```

Out-of-Module Pointer

This pattern is about pointers to addresses outside the container module range. A typical example here would be some kernel table or structure, for example, a driver IRP dispatch table having pointers outside that driver module address range. Other examples may include 32-bit SSDT pointing outside *nt* module range and IDT entries pointing outside *hal* and expected drivers:

```
[...]
818809dc 8193c4e7 nt!NtQueryOpenSubKeys
818809e0 8193c76b nt!NtQueryOpenSubKeysEx
818809e4 81a909b0 nt!NtQueryPerformanceCounter
818809e8 819920e7 nt!NtQueryQuotaInformationFile
818809ec 819e34f2 nt!NtQuerySection
818809f0 819f470b nt!NtQuerySecurityObject
818809f4 81a882fe nt!NtQuerySemaphore
818809f8 819eff54 nt!NtQuerySymbolicLinkObject
818809fc 81a8a223 nt!NtQuerySystemEnvironmentValue
81880a00 81a8a831 nt!NtQuerySystemEnvironmentValueEx
81880a04 96ca1a73
81880a08 81a7ac06 nt!NtQuerySystemTime
81880a0c 81a8f913 nt!NtQueryTimer
81880a10 81a7aeeb nt!NtQueryTimerResolution
81880a14 8193985a nt!NtQueryValueKey
81880a18 819e9273 nt!NtQueryVirtualMemory
81880a1c 8199274e nt!NtQueryVolumeInformationFile
81880a20 81a1a655 nt!NtQueueApcThread
[...]

0: kd> lm m nt
start end module name
81800000 81ba1000 nt
```

Such pointers may also be **Raw Pointers,** but it also could be the case that all pointers are raw in the absence of symbols with only a few pointing outside of the expected range.

Packed Code

This is a frequent ingredient of armored malware. Here we demonstrate a few WinDbg commands to detect UPX-packed modules with little or no expected strings:

```
0:000> !dh 00000000`00fd40b0

File Type: DLL
FILE HEADER VALUES
14C machine (i386)
3 number of sections
time date stamp Fri Jan 18 21:27:25 2013

0 file pointer to symbol table
0 number of symbols
E0 size of optional header
2102 characteristics
Executable
32 bit word machine
DLL

OPTIONAL HEADER VALUES
10B magic #
11.00 linker version
6000 size of code
1000 size of initialized data
F000 size of uninitialized data
15600 address of entry point
10000 base of code
----- new -----
0000000010000000 image base
1000 section alignment
200 file alignment
2 subsystem (Windows GUI)
6.00 operating system version
0.00 image version
6.00 subsystem version
17000 size of image
1000 size of headers
0 checksum
0000000000100000 size of stack reserve
0000000000001000 size of stack commit
0000000000100000 size of heap reserve
0000000000001000 size of heap commit
140  DLL characteristics
Dynamic base
NX compatible
16274 [      AC] address [size] of Export Directory
161DC [      98] address [size] of Import Directory
16000 [     1DC] address [size] of Resource Directory
0 [       0] address [size] of Exception Directory
0 [       0] address [size] of Security Directory
16320 [      10] address [size] of Base Relocation Directory
```

```
0 [       0] address [size] of Debug Directory
0 [       0] address [size] of Description Directory
0 [       0] address [size] of Special Directory
0 [       0] address [size] of Thread Storage Directory
157CC [      48] address [size] of Load Configuration Directory
0 [       0] address [size] of Bound Import Directory
0 [       0] address [size] of Import Address Table Directory
0 [       0] address [size] of Delay Import Directory
0 [       0] address [size] of COR20 Header Directory
0 [       0] address [size] of Reserved Directory

SECTION HEADER #1
UPX0 name
F000 virtual size
1000 virtual address
0 size of raw data
400 file pointer to raw data
0 file pointer to relocation table
0 file pointer to line numbers
0 number of relocations
0 number of line numbers
E0000080 flags
Uninitialized Data
(no align specified)
Execute Read Write

SECTION HEADER #2
UPX1 name
6000 virtual size
10000 virtual address
5A00 size of raw data
400 file pointer to raw data
0 file pointer to relocation table
0 file pointer to line numbers
0 number of relocations
0 number of line numbers
E0000040 flags
Initialized Data
(no align specified)
Execute Read Write

SECTION HEADER #3
.rsrc name
1000 virtual size
16000 virtual address
400 size of raw data
5E00 file pointer to raw data
0 file pointer to relocation table
0 file pointer to line numbers
0 number of relocations
0 number of line numbers
C0000040 flags
Initialized Data
```

```
(no align specified)
Read Write

0:000> s-sa 00000000`00fd40b0 L6600
00000000`00fd40fd  "!This program cannot be run in D"
00000000`00fd411d  "OS mode."
00000000`00fd4188  "Rich"
00000000`00fd4290  "UPX0"
00000000`00fd42b8  "UPX1"
00000000`00fd42e0  ".rsrc"
00000000`00fd448b  "3.08"
00000000`00fd4490  "UPX!_"
00000000`00fd449b  "YhHM4"
00000000`00fd44d1  "vqx"
[...]
```

Such in-memory modules (not yet initialized by a loader) can be saved to disk using **.writemem** command and unpacked. Once loaded and relocated to some address, they still have UPX sections, but they now have more strings:

```
0:000> s-sa 00000000`691c0000 L300
00000000`691c004d  "!This program cannot be run in D"
00000000`691c006d  "OS mode."
00000000`691c00d8  "Rich"
00000000`691c01e0  "UPX0"
00000000`691c0207  "`UPX1"
00000000`691c022f  "`.rsrc"
[...]
00000000`691d620b  "uGC"
00000000`691d621c  "KERNEL32.DLL"
00000000`691d622a  "LoadLibraryA"
00000000`691d6238  "GetProcAddress"
00000000`691d6248  "VirtualProtect"
00000000`691d6258  "VirtualAlloc"
00000000`691d6266  "VirtualFree"
[...]

0:000> s-su 00000000`691c0000 L(00000000`691d7000-00000000`691c0000)
[...]
00000000`691c8178  “http://www.patterndiagnostics.com”
00000000`691c8260  “mscoree.dll”
[...]
```

Patched Code

Hooksware pattern originally came from memory dump analysis pattern catalog and is too general for malware analysis pattern catalog. So we decided to factor out 3 separate patterns. The first one includes cases such as in-place patching:

```
0:004> u ntdll!ZwQueryDirectoryFile
ntdll!ZwQueryDirectoryFile:
77814db4 b8da000000      mov     eax,0DAh
77814db9 bae8af0500      mov     edx,5AFE8h
77814dbe ff12            call    dword ptr [edx]
77814dc0 c22c00          ret     2Ch
77814dc3 90              nop
ntdll!NtQueryDirectoryObject:
77814dc4 b8db000000      mov     eax,0DBh
77814dc9 ba0003fe7f      mov     edx,offset SharedUserData!SystemCallStub (7ffe0300)
77814dce ff12            call    dword ptr [edx]
```

And detour patching:

```
0:004> u wininet!InternetReadFile
wininet!InternetReadFile:
7758654b e98044ac88      jmp     0004a9d0
77586550 83ec24          sub     esp,24h
77586553 53              push    ebx
77586554 56              push    esi
77586555 57              push    edi
77586556 33ff            xor     edi,edi
77586558 393db8116277    cmp     dword ptr [wininet!GlobalDataInitialized
(776211b8)],edi
7758655e 897df4          mov     dword ptr [ebp-0Ch],edi
```

In the case of WinDbg, such a pattern is usually detected on the crash spot, such as from **RIP Stack Trace** or from the **!chkimg** command output.

Pre-Obfuscation Residue

This pattern is closely linked to packed and/or obfuscated code. Depending on a level of obfuscation and/or packing, some initial code and data structures and patterns, including fragments of strings, may leak into post-obfuscation data giving a clue to intended software behavior:

```
0:000> s-sa 00000000`00fd4000 L6000
[...]
00000000`00fd943d  "o__"
00000000`00fd9449  "91!We"
00000000`00fd945d  "H5!"
00000000`00fd94d2  "zQ@"
00000000`00fd94dd  "ommandS"
00000000`00fd94f4  "IsDeb"
00000000`00fd94fd  "uggerP"
00000000`00fd9507  "Enc"
00000000`00fd950c  "v)3Po4t"
00000000`00fd9515  "DeXU"
00000000`00fd9520  "xFe"
00000000`00fd952a  "5Eb"
00000000`00fd9533  "SI=l8kev"
00000000`00fd953e  "Z_1m"
00000000`00fd9547  "@IF"
[...]
```

Raw Pointer

This pattern is about pointers without matching symbol files. They may be in the expected module range or in some other known module range in the form of module + offset or can be completely out of range of any module from the loaded module list and, therefore, just a number. For example, usually, we have certain structures or arrays (tables) where we expect pointers with matching symbols such as IAT, IDT, and 32-bit SSDT where an occurrence of a raw pointer immediately triggers a suspicion, such as in this Import Address Table from *ProcessA*:

```
[...]
00000001`3f8a9048 00000000`76e282d0 ntdll!RtlSizeHeap
00000001`3f8a9050 00000000`76bf9070 kernel32!GetStringTypeWStub
00000001`3f8a9058 00000000`76c03580 kernel32!WideCharToMultiByteStub
00000001`3f8a9060 00000000`76e33f20 ntdll!RtlReAllocateHeap
00000001`3f8a9068 00000000`76e533a0 ntdll!RtlAllocateHeap
00000001`3f8a9070 00000000`76bfc420 kernel32!GetCommandLineWStub
00000001`3f8a9078 00000001`3f8a1638 ProcessA+0x10ac
00000001`3f8a9080 00000000`76c2cc50 kernel32!IsProcessorFeaturePresent
00000001`3f8a9088 00000000`76c02d60 kernel32!GetLastErrorStub
00000001`3f8a9090 00000000`76c02d80 kernel32!SetLastError
00000001`3f8a9098 00000000`76bf3ee0 kernel32!GetCurrentThreadIdStub
[...]
```

Note that structures are not limited to the above and can be any OS or even application-specific structure where we have symbol files. Raw pointers outside the expected module range are covered in the next pattern.

RIP Stack Trace

Injected code addresses may not be in the address ranges of loaded modules. In such cases, in the execution call history, we would see plain EIP and RIP return addresses on stack traces. We call this pattern **RIP Stack Trace** partly because we have seen these addresses after something had gone wrong and a process crashed:

```
0:005> k
ChildEBP RetAddr
02aec974 77655620 ntdll!KiFastSystemCallRet
02aec978 77683c62 ntdll!NtWaitForSingleObject+0xc
02aec9fc 77683d4b ntdll!RtlReportExceptionEx+0x14b
02aeca3c 7769fa87 ntdll!RtlReportException+0x3c
02aeca50 7769fb0d ntdll!RtlpTerminateFailureFilter+0x14
02aeca5c 775f9bdc ntdll!RtlReportCriticalFailure+0x6b
02aeca70 775f4067 ntdll!_EH4_CallFilterFunc+0x12
02aeca98 77655f79 ntdll!_except_handler4+0x8e
02aecabc 77655f4b ntdll!ExecuteHandler2+0x26
02aecb6c 77655dd7 ntdll!ExecuteHandler+0x24
02aecb6c 7769faf8 ntdll!KiUserExceptionDispatcher+0xf
02aecee0 776a0704 ntdll!RtlReportCriticalFailure+0x5b
02aecef0 776a07f2 ntdll!RtlpReportHeapFailure+0x21
02aecf24 7766b1a5 ntdll!RtlpLogHeapFailure+0xa1
02aecf6c 7765730a ntdll!RtlpCoalesceFreeBlocks+0x4b9
02aed064 77657545 ntdll!RtlpFreeHeap+0x1e2
02aed080 75e47e4b ntdll!RtlFreeHeap+0x14e
02aed0c8 77037277 kernel32!GlobalFree+0x47
02aed0dc 774b4a1f ole32!ReleaseStgMedium+0x124
02aed0f0 77517feb urlmon!ReleaseBindInfo+0x4c
02aed100 774d9a87 urlmon!CINet::ReleaseCNetObjects+0x3d
02aed118 774d93f0 urlmon!CINetHttp::OnWininetRequestHandleClosing+0x60
02aed12c 76432078 urlmon!CINet::CINetCallback+0x2de
02aed274 76438f5d wininet!InternetIndicateStatus+0xfc
02aed2a4 7643937a wininet!HANDLE_OBJECT::~HANDLE_OBJECT+0xc9
02aed2c0 7643916b
wininet!INTERNET_CONNECT_HANDLE_OBJECT::~INTERNET_CONNECT_HANDLE_OBJECT+0x209
02aed2cc 76438d5e wininet!HTTP_REQUEST_HANDLE_OBJECT::`vector deleting destructor'+0xd
02aed2dc 76434e72 wininet!HANDLE_OBJECT::Dereference+0x22
02aed2e8 76439419 wininet!DereferenceObject+0x21
02aed310 76439114 wininet!_InternetCloseHandle+0x9d
02aed330 0004aaaf wininet!InternetCloseHandle+0x11e
WARNING: Frame IP not in any known module. Following frames may be wrong.
02aed33c 774c5d25 0x4aaaf
02aed358 774c5d95 urlmon!CINet::TerminateRequest+0x82
02aed364 774c5d7c urlmon!CINet::MyUnlockRequest+0x10
02aed370 774c5d63 urlmon!CINetProtImpl::UnlockRequest+0x10
02aed37c 774c5d49 urlmon!CINetEmbdFilter::UnlockRequest+0x11
02aed388 774b743d urlmon!CINet::UnlockRequest+0x13
02aed394 774b73e1 urlmon!COInetProt::UnlockRequest+0x11
02aed3a8 774b7530 urlmon!CTransaction::UnlockRequest+0x36
02aed3b4 774b74e0 urlmon!CTransData::~CTransData+0x3a
02aed3c0 774b74c9 urlmon!CTransData::`scalar deleting destructor’+0xd
02aed3d8 774e221f urlmon!CTransData::Release+0x25
02aed3e0 774b6d0a urlmon!CReadOnlyStreamDirect::~CReadOnlyStreamDirect+0x1a
02aed3ec 774b7319 urlmon!CReadOnlyStreamDirect::`vector deleting destructor’+0xd
```

```
02aed404 774b72be urlmon!CReadOnlyStreamDirect::Release+0x25
02aed410 774b71f4 urlmon!CBinding::~CBinding+0xb9
02aed41c 774b71dd urlmon!CBinding::`scalar deleting destructor’+0xd
02aed434 6b20b0e8 urlmon!CBinding::Release+0x25
02aed448 6b20b0ba mshtml!ATL::AtlComPtrAssign+0x2b
02aed458 6b20b8de mshtml!ATL::CComPtr<IBindCallbackInternal>::operator=+0x15
02aed464 6b20b8aa mshtml!CBindingXSSFilter::TearDown+0x2b
02aed46c 6b20b887 mshtml!BindingXSSFilter_TearDown+0x19
02aed478 6b0da61a mshtml!CStreamProxy::Passivate+0x12
02aed484 6b0ddf3a mshtml!CBaseFT::Release+0x1d
02aed4ac 6b0e0b70 mshtml!CDwnBindData::TerminateBind+0x11d
02aed4b8 6b11a2a9 mshtml!CDwnBindData::TerminateOnApt+0x14
02aed4ec 6b105066 mshtml!GlobalWndOnMethodCall+0xfb
02aed50c 7742fd72 mshtml!GlobalWndProc+0x183
02aed538 7742fe4a user32!InternalCallWinProc+0x23
02aed5b0 7743018d user32!UserCallWinProcCheckWow+0x14b
02aed614 7743022b user32!DispatchMessageWorker+0x322
02aed624 6ecac1d5 user32!DispatchMessageW+0xf
02aef72c 6ec5337e ieframe!CTabWindow::_TabWindowThreadProc+0x54c
02aef7e4 760f426d ieframe!LCIETab_ThreadProc+0x2c1
02aef7f4 75e4d0e9 iertutil!CIsoScope::RegisterThread+0xab
02aef800 776319bb kernel32!BaseThreadInitThunk+0xe
02aef840 7763198e ntdll!__RtlUserThreadStart+0x23
02aef858 00000000 ntdll!_RtlUserThreadStart+0x1b
```

However, such addresses need to be checked whether they belong to .NET CLR **JIT Code** (Volume 3, page 132).

Self-Diagnosis (Kernel Mode)

This pattern is a kernel mode counterpart to **Self-Diagnosis** in user mode (Volume 2, page 318). It is just a collection of bugcheck codes where a problem is usually detected before corruption causes a fault, exception, or trap. A typical example would be a detection of a failed assertion or corrupt structures such as:

```
BAD_POOL_HEADER (19)
The pool is already corrupt at the time of the current request.
This may or may not be due to the caller.
The internal pool links must be walked to figure out a possible cause of the problem,
and then special pool applied to the suspect tags or the driver verifier to a suspect
driver.
Arguments:
Arg1: 00000020, a pool block header size is corrupt.
Arg2: 8b79d078, The pool entry we were looking for within the page.
Arg3: 8b79d158, The next pool entry.
Arg4: 8a1c0004, (reserved)
```

Stack Trace Collection

Sometimes a problem can be identified not from a single **Stack Trace** pattern but a **Stack Trace Collection**.

These include **Coupled Processes (**Volume 1, page 419), **Procedure Call Chains (**Volume 1, page 482**),** and **Blocked Threads (**Volume 2**)**. Here I only discuss various methods to list stack traces.

- Process dumps including various process minidumps:

~*kv command lists all process threads.

!findstack *module[!symbol]* **2** command filters all stack traces to show ones containing *module* or *module!symbol.*

!uniqstack command.

- Kernel minidumps:

have only one problem thread. The **kv** command or its variant is sufficient.

- Kernel and complete memory dumps:

!process 0 3f command lists all processes and their threads, including user space process thread stacks for complete memory dumps. This command is valid for Windows XP and later. For older systems, we can use WinDbg scripts.

!stacks 2 *[module[!symbol]]* command shows kernel mode stack traces, and we can filter the output based on *module* or *module!symbol.* Filtering is valid only for crash dumps from Windows XP and later systems.

~*[ProcessorN]***s;.reload /user;kv** command sequence shows the stack trace for the running thread on the specified processor.

The processor change command is illustrated in this example:

```
0: kd> ~2s

2: kd> k
ChildEBP RetAddr
eb42bd58 00000000 nt!KiIdleLoop+0x14

2: kd> ~1s;.reload /user;k
Loading User Symbols
...
ChildEBP RetAddr
be4f8c30 eb091f43 i8042prt!I8xProcessCrashDump+0x53
be4f8c8c 8046bfe2 i8042prt!I8042KeyboardInterruptService+0x15d
be4f8c8c 8049470f nt!KiInterruptDispatch+0x32
be4f8d54 80468389 nt!NtSetEvent+0x71
be4f8d54 77f8290a nt!KiSystemService+0xc9
081cfefc 77f88266 ntdll!ZwSetEvent+0xb
081cff0c 77f881b1 ntdll!RtlpUnWaitCriticalSection+0x1b
081cff14 1b00c7d1 ntdll!RtlLeaveCriticalSection+0x1d
```

```
081cff4c 1b0034da msjet40!Database::ReadPages+0x81
081cffb4 7c57b3bc msjet40!System::WorkerThread+0x115
081cffec 00000000 KERNEL32!BaseThreadStart+0x52
```

Example of **!findstack** command (process dump):

```
0:000> !findstack kernel32!RaiseException 2
Thread 000, 1 frame(s) match
* 00 0013b3f8 72e8d3ef kernel32!RaiseException+0x53
  01 0013b418 72e9a26b msxml3!Exception::raiseException+0x5f
  02 0013b424 72e8ff00 msxml3!Exception::_throwError+0x22
  03 0013b46c 72e6abaa msxml3!COMSafeControlRoot::getBaseURL+0x3d
  04 0013b4bc 72e6a888 msxml3!Document::loadXML+0x82
  05 0013b510 64b73a9b msxml3!DOMDocumentWrapper::loadXML+0x5a
  06 0013b538 64b74eb6 iepeers!CPersistUserData::initXMLCache+0xa6
  07 0013b560 77d0516e iepeers!CPersistUserData::load+0xfc
  08 0013b57c 77d14abf oleaut32!DispCallFunc+0x16a
...
...
...
  66 0013fec8 0040243d shdocvw!IEWinMain+0x129
  67 0013ff1c 00402744 iexplore!WinMain+0x316
  68 0013ffc0 77e6f23b iexplore!WinMainCRTStartup+0x182
  69 0013fff0 00000000 kernel32!BaseProcessStart+0x23
```

Example of **!stacks** command (kernel dump):

```
2: kd> !stacks 2 nt!PspExitThread
Proc.Thread  .Thread  Ticks   ThreadState Blocker
                            [8a390818 System]

                            [8a1bbbf8 smss.exe]

                            [8a16cbf8 csrss.exe]

                            [89c14bf0 winlogon.exe]

                            [89dda630 services.exe]

                            [89c23af0 lsass.exe]

                            [8a227470 svchost.exe]

                            [89f03bb8 svchost.exe]

                            [89de3820 svchost.exe]

                            [89d09b60 svchost.exe]

                            [89c03530 ccEvtMgr.exe]

                            [89b8f4f0 ccSetMgr.exe]

                            [89dfe8c0 SPBBCSvc.exe]

                            [89c9db18 svchost.exe]

                            [89dfa268 spoolsv.exe]
```

```
[89dfa6b8 msdtc.exe]

[89df38f0 CpSvc.exe]

[89d97d88 DefWatch.exe]

[89e04020 IBMSPSVC.EXE]

[89b54710 IBMSPREM.EXE]

[89d9e4b0 IBMSPREM.EXE]

[89c2c4e8 svchost.exe]

[89d307c0 SavRoam.exe]

[89bfcd88 Rtvscan.exe]

[89b53b60 uphclean.exe]

[89c24020 AgentSVC.exe]

[89d75b60 sAginst.exe]

[89cf0d88 CdfSvc.exe]

[89d87020 cdmsvc.exe]

[89dafd88 ctxxmlss.exe]

[89d8dd88 encsvc.exe]

[89d06d88 ImaSrv.exe]

[89d37b60 mfcom.exe]

[89c8bb18 SmaService.exe]

[89d2ba80 svchost.exe]

[89ce8630 XTE.exe]

[89b64b60 XTE.exe]

[89b7c680 ctxcpusched.exe]

[88d94a88 ctxcpuusync.exe]

[89ba5418 unsecapp.exe]

[89d846e0 wmiprvse.exe]

[89cda9d8 ctxwmisvc.exe]

[88d6cb78 logon.scr]
```

```
                           [88ba0a70 csrss.exe]

                           [88961968 winlogon.exe]

                           [8865f740 rdpclip.exe]

                           [8858db20 wfshell.exe]

                           [88754020 explorer.exe]

                           [88846d88 BacsTray.exe]

                           [886b6180 ccApp.exe]

                           [884bc020 fppdis3a.exe]

                           [885cb350 ctfmon.exe]

                           [888bb918 cscript.exe]

                           [8880b3c8 cscript.exe]

                           [88ad2950 csrss.exe]
b68.00215c 88930020 0000000 RUNNING nt!KeBugCheckEx+0x1b
                                    nt!MiCheckSessionPoolAllocations+0xe3
                                    nt!MiDereferenceSessionFinal+0x183
                                    nt!MmCleanProcessAddressSpace+0x6b
                                    nt!PspExitThread+0x5f1
                                    nt!PspTerminateThreadByPointer+0x4b
                                    nt!PspSystemThreadStartup+0x3c
                                    nt!KiThreadStartup+0x16

                           [88629310 winlogon.exe]

                           [88a4d9b0 csrss.exe]

                           [88d9f8b0 winlogon.exe]

                           [88cd5840 wfshell.exe]

                           [8a252440 OUTLOOK.EXE]

                           [8a194bf8 WINWORD.EXE]

                           [88aabd20 ctfmon.exe]

                           [889ef440 EXCEL.EXE]

                           [88bec838 HogiaGUI2.exe]

                           [88692020 csrss.exe]

                           [884dd508 winlogon.exe]

                           [88be1d88 wfshell.exe]

                           [886a7d88 OUTLOOK.EXE]
```

```
                            [889baa70 WINWORD.EXE]

                            [8861e3d0 ctfmon.exe]

                            [887bbb68 EXCEL.EXE]

                            [884e4020 csrss.exe]

                            [8889d218 winlogon.exe]

                            [887c8020 wfshell.exe]

Threads Processed: 1101
```

What if we have a list of processes from a complete memory dump by using **!process 0 0** command and we want to interrogate the specific process? In this case, we need to switch to that process and reload user space symbol files (**.process /r /p** *address)*.

There is also a separate command to reload user space symbol files any time (**.reload /user)**.

After switching, we can list threads (**!process** *address*), dump or search process virtual memory. For example:

```
1: kd> !process 0 0
**** NT ACTIVE PROCESS DUMP ****
PROCESS 890a3320  SessionId: 0  Cid: 0008    Peb: 00000000  ParentCid: 0000
    DirBase: 00030000  ObjectTable: 890a3e08  TableSize: 405.
    Image: System

PROCESS 889dfd60  SessionId: 0  Cid: 0144    Peb: 7ffdf000  ParentCid: 0008
    DirBase: 0b9e7000  ObjectTable: 889fdb48  TableSize: 212.
    Image: SMSS.EXE

PROCESS 890af020  SessionId: 0  Cid: 0160    Peb: 7ffdf000  ParentCid: 0144
    DirBase: 0ce36000  ObjectTable: 8898e308  TableSize: 747.
    Image: CSRSS.EXE

PROCESS 8893d020  SessionId: 0  Cid: 0178    Peb: 7ffdf000  ParentCid: 0144
    DirBase: 0d33b000  ObjectTable: 890ab4c8  TableSize: 364.
    Image: WINLOGON.EXE

PROCESS 88936020  SessionId: 0  Cid: 0194    Peb: 7ffdf000  ParentCid: 0178
    DirBase: 0d7d5000  ObjectTable: 88980528  TableSize: 872.
    Image: SERVICES.EXE

PROCESS 8897f020  SessionId: 0  Cid: 01a0    Peb: 7ffdf000  ParentCid: 0178
    DirBase: 0d89d000  ObjectTable: 889367c8  TableSize: 623.
    Image: LSASS.EXE

1: kd> .process /r /p 8893d020
Implicit process is now 8893d020
Loading User Symbols
...

1: kd> !process 8893d020
PROCESS 8893d020  SessionId: 0  Cid: 0178    Peb: 7ffdf000  ParentCid: 0144
    DirBase: 0d33b000  ObjectTable: 890ab4c8  TableSize: 364.
    Image: WINLOGON.EXE
    VadRoot 8893a508 Clone 0 Private 1320. Modified 45178. Locked 0.
    DeviceMap 89072448
    Token                             e392f8d0
    ElapsedTime                        9:54:06.0882
    UserTime                          0:00:00.0071
    KernelTime                        0:00:00.0382
    QuotaPoolUsage[PagedPool]         34828
    QuotaPoolUsage[NonPagedPool]      43440
    Working Set Sizes (now,min,max)  (737, 50, 345) (2948KB, 200KB, 1380KB)
    PeakWorkingSetSize                2764
    VirtualSize                       46 Mb
    PeakVirtualSize                   52 Mb
    PageFaultCount                    117462
    MemoryPriority                    FOREGROUND
    BasePriority                      13
    CommitCharge                      1861

        THREAD 8893dda0  Cid 178.15c  Teb: 7ffde000  Win32Thread: a2034908 WAIT:
(WrUserRequest) UserMode Non-Alertable
            8893bee0  SynchronizationEvent
        Not impersonating
        Owning Process 8893d020
        Wait Start TickCount    29932455      Elapsed Ticks: 7
        Context Switch Count    28087                   LargeStack
        UserTime                  0:00:00.0023
        KernelTime                0:00:00.0084
        Start Address winlogon!WinMainCRTStartup (0x0101cbb0)
        Stack Init eb1b0000 Current eb1afcc8 Base eb1b0000 Limit eb1ac000 Call 0
        Priority 15 BasePriority 15 PriorityDecrement 0 DecrementCount 0

        ChildEBP RetAddr
        eb1afce0 8042d893 nt!KiSwapThread+0x1b1
        eb1afd08 a00019c2 nt!KeWaitForSingleObject+0x1a3
        eb1afd44 a0013993 win32k!xxxSleepThread+0x18a
        eb1afd54 a001399f win32k!xxxWaitMessage+0xe
        eb1afd5c 80468389 win32k!NtUserWaitMessage+0xb
        eb1afd5c 77e58b53 nt!KiSystemService+0xc9
        0006fdd0 77e33630 USER32!NtUserWaitMessage+0xb
        0006fe04 77e44327 USER32!DialogBox2+0x216
        0006fe28 77e38d37 USER32!InternalDialogBox+0xd1
        0006fe48 77e39eba USER32!DialogBoxIndirectParamAorW+0x34
        0006fe6c 01011749 USER32!DialogBoxParamW+0x3d
        0006fea8 01018bd3 winlogon!TimeoutDialogBoxParam+0x27
        0006fee0 76b93701 winlogon!WlxDialogBoxParam+0x7b
        0006ff08 010164c6 3rdPartyGINA!WlxDisplaySASNotice+0x43
        0006ff20 01014960 winlogon!MainLoop+0x96
        0006ff58 0101cd06 winlogon!WinMain+0x37a
        0006fff4 00000000 winlogon!WinMainCRTStartup+0x156
```

```
THREAD 88980020  Cid 178.188  Teb: 7ffdc000  Win32Thread: 00000000 WAIT:
(DelayExecution) UserMode Alertable
            88980108  NotificationTimer
        Not impersonating
        Owning Process 8893d020
        Wait Start TickCount    29930810      Elapsed Ticks: 1652
        Context Switch Count    15638
        UserTime                  0:00:00.0000
        KernelTime                0:00:00.0000
        Start Address KERNEL32!BaseThreadStartThunk (0x7c57b740)
        Win32 Start Address ntdll!RtlpTimerThread (0x77faa02d)
        Stack Init bf6f7000 Current bf6f6cc4 Base bf6f7000 Limit bf6f4000 Call 0
        Priority 13 BasePriority 13 PriorityDecrement 0 DecrementCount 0

        ChildEBP RetAddr
        bf6f6cdc 8042d340 nt!KiSwapThread+0x1b1
        bf6f6d04 8052aac9 nt!KeDelayExecutionThread+0x182
        bf6f6d54 80468389 nt!NtDelayExecution+0x7f
        bf6f6d54 77f82831 nt!KiSystemService+0xc9
        00bfff9c 77f842c4 ntdll!NtDelayExecution+0xb
        00bfffb4 7c57b3bc ntdll!RtlpTimerThread+0x42
        00bfffec 00000000 KERNEL32!BaseThreadStart+0x52

1: kd> dds 0006fee0
0006fee0  0006ff08
0006fee4  76b93701 3rdPartyGINA!WlxDisplaySASNotice+0x43
0006fee8  000755e8
0006feec  76b90000 3rdParty
0006fef0  00000578
0006fef4  00000000
0006fef8  76b9370b 3rdParty!WlxDisplaySASNotice+0x4d
0006fefc  0008d0e0
0006ff00  00000008
0006ff04  00000080
0006ff08  0006ff20
0006ff0c  010164c6 winlogon!MainLoop+0x96
0006ff10  0008d0e0
0006ff14  5ffa0000
0006ff18  000755e8
0006ff1c  00000000
0006ff20  0006ff58
0006ff24  01014960 winlogon!WinMain+0x37a
0006ff28  000755e8
0006ff2c  00000005
0006ff30  00072c9c
0006ff34  00000001
0006ff38  000001bc
0006ff3c  00000005
0006ff40  00000001
0006ff44  0000000d
0006ff48  00000000
0006ff4c  00000000
0006ff50  00000000
0006ff54  0000ffe4
0006ff58  0006fff4
0006ff5c  0101cd06 winlogon!WinMainCRTStartup+0x156
```

We can also filter stacks that belong to processes having the same module name, for example, **svchost.exe** (see **Filtering Processes**, Volume 1, page 220).

Sometimes the collection of all stack traces from all threads in the system can disprove or decrease the plausibility of the hypothesis that some module is involved. In one case, the customer claimed that the specific driver was involved in the server freeze. However, there was no such module found in all thread stacks.

Stack Trace Collection (I/O Requests)

In addition to stack trace collections for threads (unmanaged, Volume 1, page 409, managed, Volume 6, page 127, and predicate, Volume 7, page 100), we introduce an additional pattern for I/O requests. Such requests are implemented via the so-called I/O request packets (IRP) that “travel” from a device driver to a device driver similar to a C++ class method to another C++ class method (where a device object address is similar to a C++ object instance address). An IRP stack is used to keep track of the current driver processing an IRP that is reused between device drivers. It is basically an array of structures describing how a particular driver function was called with appropriate parameters similar to a call frame on an execution thread stack. A long time ago, we created a UML diagram depicting the flow of an IRP through the driver (device) stack (diagram #3, Volume 1, page 700). An I/O stack location pointer is decremented (from the bottom to the top) as a thread stack pointer (ESP or RSP). We can list active and completed I/O requests with their stack traces using the **!irpfind -v** WinDbg command:

```
1: kd> !irpfind -v

Scanning large pool allocation table for Tag: Irp? (832c7000 : 833c7000)

Irp    [ Thread ] irpStack: (Mj,Mn)   DevObj  [Driver]         MDL Process
8883dc18: Irp is active with 1 stacks 1 is current (= 0x8883dc88)
 No Mdl: No System Buffer: Thread 888f8950:  Irp stack trace.
      cmd  flg cl Device   File     Completion-Context
> [  d, 0]   5  1 88515ae8 888f82f0 00000000-00000000    pending
             \FileSystem\Npfs
                   Args: 00000000 00000000 00110008 00000000

891204c8: Irp is active with 1 stacks 1 is current (= 0x89120538)
 No Mdl: No System Buffer: Thread 889635b0:  Irp stack trace.
      cmd  flg cl Device   File     Completion-Context
> [  3, 0]   0  1 88515ae8 84752028 00000000-00000000    pending
             \FileSystem\Npfs
                   Args: 0000022a 00000000 00000000 00000000

89120ce8: Irp is active with 1 stacks 1 is current (= 0x89120d58)
 No Mdl: No System Buffer: Thread 89212030:  Irp stack trace.
      cmd  flg cl Device   File     Completion-Context
> [  3, 0]   0  1 88515ae8 8921be00 00000000-00000000    pending
             \FileSystem\Npfs
                   Args: 0000022a 00000000 00000000 00000000

Searching NonPaged pool (80000000 : ffc00000) for Tag: Irp?

[...]

892cbe48: Irp is active with 9 stacks 9 is current (= 0x892cbfd8)
 No Mdl: No System Buffer: Thread 892add78:  Irp stack trace.
    cmd  flg cl Device   File     Completion-Context
[  0, 0]   0  0 00000000 00000000 00000000-00000000

                 Args: 00000000 00000000 00000000 00000000
[  0, 0]   0  0 00000000 00000000 00000000-00000000
```

```
                 Args: 00000000 00000000 00000000 00000000
[  0, 0]   0  0 00000000 00000000 00000000-00000000

                 Args: 00000000 00000000 00000000 00000000
[  0, 0]   0  0 00000000 00000000 00000000-00000000

                 Args: 00000000 00000000 00000000 00000000
[  0, 0]   0  0 00000000 00000000 00000000-00000000

                 Args: 00000000 00000000 00000000 00000000
[  0, 0]   0  0 00000000 00000000 00000000-00000000

                 Args: 00000000 00000000 00000000 00000000
[  0, 0]   0  0 00000000 00000000 00000000-00000000

                 Args: 00000000 00000000 00000000 00000000
[  0, 0]   0  0 00000000 00000000 00000000-00000000

                 Args: 00000000 00000000 00000000 00000000
> [  c, 2]   0  1 8474a020 892c8c80 00000000-00000000    pending
             \FileSystem\Ntfs
                 Args: 00000800 00000002 00000000 00000000

892daa88: Irp is active with 4 stacks 4 is current (= 0x892dab64)
 No Mdl: System buffer=831559c8: Thread 8322c8e8:  Irp stack trace.
    cmd  flg cl Device   File     Completion-Context
[  0, 0]   0  0 00000000 00000000 00000000-00000000

                 Args: 00000000 00000000 00000000 00000000
[  0, 0]   0  0 00000000 00000000 00000000-00000000

                 Args: 00000000 00000000 00000000 00000000
[  0, 0]   0  0 00000000 00000000 00000000-00000000

                 Args: 00000000 00000000 00000000 00000000
> [  e,2d]   5  1 884ba750 83190c40 00000000-00000000    pending
             \Driver\AFD
                 Args: 890cbc44 890cbc44 88e55297 8943b6c8

892ea4e8: Irp is active with 4 stacks 4 is current (= 0x892ea5c4)
 No Mdl: No System Buffer: Thread 00000000:  Irp stack trace.  Pending has been
returned
    cmd  flg cl Device   File     Completion-Context
[  0, 0]   0  2 00000000 00000000 00000000-00000000

                 Args: 00000000 00000000 00000000 c0000185
[  0, 0]   0  0 00000000 00000000 00000000-00000000

                 Args: 00000000 00000000 00000000 00000000
[  f, 0]   0  2 83a34bb0 00000000 84d779ed-88958050
              \Driver\atapi CLASSPNP!ClasspMediaChangeDetectionCompletion
                 Args: 88958050 00000000 00000000 83992d10
> [  0, 0]   2  0 891ee030 00000000 00000000-00000000
```

```
              \Driver\cdrom
                 Args: 00000000 00000000 00000000 00000000

8933fcb0: Irp is active with 1 stacks 1 is current (= 0x8933fd20)
 No Mdl: No System Buffer: Thread 84753d78:  Irp stack trace.
      cmd  flg cl Device   File     Completion-Context
> [  3, 0]   0  1 88515ae8 84759f40 00000000-00000000    pending
               \FileSystem\Npfs
                   Args: 0000022a 00000000 00000000 00000000

893cf550: Irp is active with 1 stacks 1 is current (= 0x893cf5c0)
 No Mdl: No System Buffer: Thread 888fd3b8:  Irp stack trace.
      cmd  flg cl Device   File     Completion-Context
> [  3, 0]   0  1 88515ae8 834d30d0 00000000-00000000    pending
               \FileSystem\Npfs
                   Args: 00000400 00000000 00000000 00000000

893da468: Irp is active with 6 stacks 7 is current (= 0x893da5b0)
 Mdl=892878f0: No System Buffer: Thread 00000000:  Irp is completed.  Pending has been
returned
    cmd  flg cl Device   File     Completion-Context
[  0, 0]   0  0 00000000 00000000 00000000-00000000

                 Args: 00000000 00000000 00000000 00000000
[  0, 0]   0  0 00000000 00000000 00000000-00000000

                 Args: 00000000 00000000 00000000 00000000
[  0, 0]   0  0 00000000 00000000 00000000-00000000

                 Args: 00000000 00000000 00000000 00000000
[  0, 0]   0  0 00000000 00000000 00000000-00000000

                 Args: 00000000 00000000 00000000 00000000
[  f, 0]   0  0 84b3e028 00000000 9747fcd0-00000000
              \Driver\usbehci USBSTOR!USBSTOR_CswCompletion
                 Args: 00000000 00000000 00000000 00000000
[  f, 0]   0  0 892ba8f8 00000000 84d780ce-8328e0f0
              \Driver\USBSTOR CLASSPNP!TransferPktComplete
                 Args: 00000000 00000000 00000000 00000000

893efb00: Irp is active with 10 stacks 11 is current (= 0x893efcd8)
 Mdl=83159378: No System Buffer: Thread 82b7f828:  Irp is completed.  Pending has been
returned
    cmd  flg cl Device   File     Completion-Context
[  0, 0]   0  0 00000000 00000000 00000000-00000000

                 Args: 00000000 00000000 00000000 00000000
[  0, 0]   0  0 00000000 00000000 00000000-00000000

                 Args: 00000000 00000000 00000000 00000000
[  0, 0]   0  0 00000000 00000000 00000000-00000000
```

```
                 Args: 00000000 00000000 00000000 00000000
[  0, 0]   0  0 00000000 00000000 00000000-00000000

                 Args: 00000000 00000000 00000000 00000000
[  0, 0]   0  0 00000000 00000000 00000000-00000000

                 Args: 00000000 00000000 00000000 00000000
[  0, 0]   0  0 00000000 00000000 00000000-00000000

                 Args: 00000000 00000000 00000000 00000000
[  3, 0]   0  0 885a55b8 00000000 81614138-00000000
             \Driver\disk partmgr!PmReadWriteCompletion
                 Args: 00000000 00000000 00000000 00000000
[  3, 0]   0  0 89257c90 00000000 8042e4d4-831caab0
             \Driver\partmgr volmgr!VmpReadWriteCompletionRoutine
                 Args: 00000000 00000000 00000000 00000000
[  3, 0]   0  0 831ca9f8 00000000 84dad0be-00000000
             \Driver\volmgr ecache!EcDispatchReadWriteCompletion
                 Args: 00000000 00000000 00000000 00000000
[  3, 0]   0  0 8319c020 00000000 84dcc4d4-8576f8ac
             \Driver\Ecache volsnap!VspSignalCompletion
                 Args: 00000000 00000000 00000000 00000000
```

String Hint

This pattern covers traces of ASCII and UNICODE strings that look suspicious such as website, password, and HTTP forms or strange names that intuitively shouldn’t be present according to the purpose of a module or its container process:

```
0:005> s-sa 00040000 L1d000
0004004d  "!This program cannot be run in D"
0004006d  "OS mode."
00040081  "3y@"
000400b8  "Rich"
000401d0  ".text"
000401f7  "`.rdata"
0004021f  "@.data"
00040248  ".reloc"
[...]
00054018  "GET /stat?uptime=%d&downlink=%d&"
00054038  "uplink=%d&id=%s&statpass=%s&comm"
00054058  "ent=%s HTTP/1.0"
000540ac  "%s%s%s"
000540d8  "ftp://%s:%s@%s:%d"
000540fc  "Accept-Encoding:"
00054118  "Accept-Encoding:"
00054130  "0123456789ABCDEF"
00054144  "://"
00054160  "POST %s HTTP/1.0"
00054172  "Host: %s"
0005417c  "User-Agent: %s"
0005418c  "Accept: text/html"
0005419f  "Connection: Close"
000541b2  "Content-Type: application/x-www-"
000541d2  "form-urlencoded"
000541e3  "Content-Length: %d"
000541fc  "id="
00054208  "POST %s HTTP/1.1"
0005421a  "Host: %s"
00054224  "User-Agent: %s"
00054234  "Accept: text/html"
00054247  "Connection: Close"
0005425a  "Content-Type: application/x-www-"
0005427a  "form-urlencoded"
0005428b  "Content-Length: %d"
000542a4  "id=%s&base="
000542b8  "id=%s&brw=%d&type=%d&data="
000542d8  "POST %s HTTP/1.1"
000542ea  "Host: %s"
000542f4  "User-Agent: %s"
00054304  "Accept: text/html"
00054317  "Connection: Close"
0005432a  "Content-Type: application/x-www-"
0005434a  "form-urlencoded"
0005435b  "Content-Length: %d"
00054378  "id=%s&os=%s&plist="
00054390  "POST %s HTTP/1.1"
```

```
000543a2  "Host: %s"
000543ac  "User-Agent: %s"
000543bc  "Accept: text/html"
000543cf  "Connection: Close"
000543e2  "Content-Type: application/x-www-"
00054402  "form-urlencoded"
00054413  "Content-Length: %d"
00054430  "id=%s&data=%s"
00054440  "POST %s HTTP/1.1"
00054452  "Host: %s"
0005445c  "User-Agent: %s"
0005446c  "Accept: text/html"
0005447f  "Connection: Close"
00054492  "Content-Type: application/x-www-"
000544b2  "form-urlencoded"
000544c3  "Content-Length: %d"
000544e0  "GET %s HTTP/1.0"
000544f1  "Host: %s"
000544fb  "User-Agent: %s"
0005450b  "Connection: close"
00054528  "POST /get/scr.html HTTP/1.0"
00054545  "Host: %s"
0005454f  "User-Agent: %s"
0005455f  "Connection: close"
00054572  "Content-Length: %d"
00054586  "Content-Type: multipart/form-dat"
000545a6  "a; boundary=--------------------"
000545c6  "-------%d"
000545d4  "-----------------------------%d"
000545f8  "%sContent-Disposition: form-data"
00054618  "; name="id""
00054630  "%sContent-Disposition: form-data"
00054650  "; name="screen"; filename="%d""
00054670  "Content-Type: application/octet-"
00054690  "stream"
000546a0  "%s(%d) : %s"
000546ac  "%s failed with error %d: %s"
000546c8  "%02X"
000546d8  "BlackwoodPRO"
000546e8  "FinamDirect"
000546f4  "GrayBox"
000546fc  "MbtPRO"
00054704  "Laser"
0005470c  "LightSpeed"
00054718  "LTGroup"
00054720  "Mbt"
00054724  "ScotTrader"
00054730  "SaxoTrader"
00054740  "Program:   %s"
0005474f  "Username:  %s"
0005475e  "Password:  %s"
0005476d  "AccountNO: %s"
[...]
```

Unknown Module

Sometimes we suspect a problem was caused by some module, but the WinDbg **lmv** command doesn't show the company name and other verbose information for it, and Google search has no results for the file name. We call this pattern **Unknown Component (Module)**.

In such cases, additional information can be obtained by dumping the module resource section or the whole module address range and looking for ASCII and UNICODE strings. For example (byte values in the **db** output are omitted for clarity):

```
2: kd> lmv m driver
start    end        module name
f5022000 f503e400   driver   (deferred)
    Image path: \SystemRoot\System32\drivers\driver.sys
    Image name: driver.sys
    Timestamp:        Tue Jun 12 11:33:16 2007 (466E766C)
    CheckSum:         00021A2C
    ImageSize:        0001C400
    Translations:     0000.04b0 0000.04e0 0409.04b0 0409.04e0

2: kd> db f5022000 f503e400
f5022000  MZ..............
f5022010  ........@.......
f5022020  ................
f5022030  ................
f5022040  ........!..L.!Th
f5022050  is program canno
f5022060  t be run in DOS 
f5022070  mode....$.......
f5022080  .g,._.B._.B._.B.
f5022090  _.C.=.B..%Q.X.B.
f50220a0  _.B.].B.Y%H.|.B.
f50220b0  ..D.^.B.Rich_.B.
f50220c0  ........PE..L...
f50220d0  lvnF............
...
...
...
f503ce30  ................
f503ce40  ................
f503ce50  ................
f503ce60  ............0...
f503ce70  ................
f503ce80  ....H...........
f503ce90  ..........4...V.
f503cea0  S._.V.E.R.S.I.O.
f503ceb0  N._.I.N.F.O.....
f503cec0  ................
f503ced0  ........?.......
f503cee0  ................
f503cef0  ....P.....S.t.r.
f503cf00  i.n.g.F.i.l.e.I.
f503cf10  n.f.o...,.....0.
f503cf20  4.0.9.0.4.b.0...
f503cf30  4.....C.o.m.p.a.
f503cf40  n.y.N.a.m.e.....
f503cf50  M.y.C.o.m.p. .A.
f503cf60  G...p.$...F.i.l.
f503cf70  e.D.e.s.c.r.i.p.
f503cf80  t.i.o.n.....M.y.
```

```
f503cf90  .B.i.g. .P.r.o.
f503cfa0  d.u.c.t. .H.o.o.
f503cfb0  k...............
f503cfc0  ................
f503cfd0  ....4.....F.i.l.
f503cfe0  e.V.e.r.s.i.o.n.
f503cff0  ....5...1...0...
f503d000  ????????????????
f503d010  ????????????????
f503d020  ????????????????
f503d030  ????????????????
...
...
...
```

We see that *CompanyName* is “MyComp AG”, *FileDescription* is “My Big Product Hook”, and *FileVersion* is “5.0.1”.

In our example, the same information can be retrieved by dumping the image file header and then finding and dumping the resource section:

```
2: kd> lmv m driver
start    end        module name
f5022000 f503e400   driver   (deferred)
    Image path: \SystemRoot\System32\drivers\driver.sys
    Image name: driver.sys
    Timestamp:        Tue Jun 12 11:33:16 2007 (466E766C)
    CheckSum:         00021A2C
    ImageSize:        0001C400
    Translations:     0000.04b0 0000.04e0 0409.04b0 0409.04e0

2: kd> !dh f5022000 -f

File Type: EXECUTABLE IMAGE
FILE HEADER VALUES
     14C machine (i386)
       6 number of sections
466E766C time date stamp Tue Jun 12 11:33:16 2007

       0 file pointer to symbol table
       0 number of symbols
      E0 size of optional header
     10E characteristics
            Executable
            Line numbers stripped
            Symbols stripped
            32 bit word machine

OPTIONAL HEADER VALUES
     10B magic #
    6.00 linker version
   190A0 size of code
    30A0 size of initialized data
       0 size of uninitialized data
   1A340 address of entry point
     2C0 base of code
         ----- new -----
00010000 image base
      20 section alignment
      20 file alignment
```

```
       1 subsystem (Native)
    4.00 operating system version
    0.00 image version
    4.00 subsystem version
   1C400 size of image
     2C0 size of headers
   21A2C checksum
00100000 size of stack reserve
00001000 size of stack commit
00100000 size of heap reserve
00001000 size of heap commit
       0 [       0] address [size] of Export Directory
   1A580 [      50] address [size] of Import Directory
   1AE40 [     348] address [size] of Resource Directory
       0 [       0] address [size] of Exception Directory
       0 [       0] address [size] of Security Directory
   1B1A0 [    1084] address [size] of Base Relocation Directory
     420 [      1C] address [size] of Debug Directory
       0 [       0] address [size] of Description Directory
       0 [       0] address [size] of Special Directory
       0 [       0] address [size] of Thread Storage Directory
       0 [       0] address [size] of Load Configuration Directory
       0 [       0] address [size] of Bound Import Directory
     2C0 [     15C] address [size] of Import Address Table Directory
       0 [       0] address [size] of Delay Import Directory
       0 [       0] address [size] of COR20 Header Directory
       0 [       0] address [size] of Reserved Directory

2: kd> db f5022000+1AE40 f5022000+1AE40+348
f503ce40  ................
f503ce50  ................
f503ce60  ............0...
f503ce70  ................
f503ce80  ....H...........
f503ce90  ..........4...V.
f503cea0  S._.V.E.R.S.I.O.
f503ceb0  N._.I.N.F.O.....
f503cec0  ................
f503ced0  ........?.......
f503cee0  ................
f503cef0  ....P.....S.t.r.
f503cf00  i.n.g.F.i.l.e.I.
f503cf10  n.f.o...,.....0.
f503cf20  4.0.9.0.4.b.0...
f503cf30  4.....C.o.m.p.a.
f503cf40  n.y.N.a.m.e.....
f503cf50  M.y.C.o.m.p. .A.
f503cf60  G...p.$...F.i.l.
f503cf70  e.D.e.s.c.r.i.p.
f503cf80  t.i.o.n.....M.y.
f503cf90  .B.i.g. .P.r.o.
f503cfa0  d.u.c.t. .H.o.o.
f503cfb0  k...............
f503cfc0  ................
f503cfd0  ....4.....F.i.l.
f503cfe0  e.V.e.r.s.i.o.n.
f503cff0  ....5...1...0...
f503d000  ????????????????
f503d010  ????????????????
...
...
...
```

Raw Stack Dump of All Threads (Kernel Space)

Having done in the past with user space raw stack data analysis for 32-bit complete memory dumps (Volume 1, page 236) we found today the need to look at kernel raw stack data from all threads and created this fast script:

```
!for_each_thread "!thread @#Thread; r? $t1 = ((nt!_KTHREAD *) @#Thread )->StackLimit;
r? $t2 = ((nt!_KTHREAD *) @#Thread )->InitialStack; dps @$t1 @$t2"
```

It can be run for kernel and complete memory dumps from both x86 and x64 systems. If we need correct symbolic mapping for user space in kernel space data, we need to modify it a bit, and it is slower to run.

```
!for_each_thread "!thread @#Thread 3f; .thread /r /p @#Thread; r? $t1 = ((nt!_KTHREAD
*) @#Thread )->StackLimit; r? $t2 = ((nt!_KTHREAD *) @#Thread )->InitialStack; dps @$t1
@$t2"
```

Complete Stack Traces from x64 System

Previously we wrote about how to get a 32-bit stack trace from a 32-bit process thread on an x64 system (Volume 3, page 43). There are situations when we are interested in all such stack traces, for example, from a complete memory dump. We wrote a script that extracted both 64-bit and WOW64 32-bit stack traces:

```
.load wow64exts
!for_each_thread "!thread @#Thread 1f;.thread /w @#Thread; .reload; kb 256; .effmach
AMD64"
```

Here is WinDbg example output fragment for a thread fffffa801f3a3bb0 from a very long debugger log file:

```
[...]

Setting context for owner process...
.process /p /r fffffa8013177c10


THREAD fffffa801f3a3bb0  Cid 4b4c.5fec  Teb: 000000007efaa000 Win32Thread: fffff900c1efad50 WAIT:
(UserRequest) UserMode Non-Alertable
    fffffa8021ce4590  NotificationEvent
    fffffa801f3a3c68  NotificationTimer
Not impersonating
DeviceMap                 fffff8801b551720
Owning Process            fffffa8013177c10       Image:         application.exe
Attached Process          N/A            Image:         N/A
Wait Start TickCount      14066428       Ticks: 301 (0:00:00:04.695)
Context Switch Count      248                 LargeStack
UserTime                  00:00:00.000
KernelTime                00:00:00.000
Win32 Start Address mscorwks!Thread::intermediateThreadProc (0x00000000733853b3)
Stack Init fffffa60190e5db0 Current fffffa60190e5940
Base fffffa60190e6000 Limit fffffa60190df000 Call 0
Priority 11 BasePriority 10 PriorityDecrement 0 IoPriority 2 PagePriority 5
Child-SP          RetAddr           Call Site
fffffa60`190e5980 fffff800`01cba0fa nt!KiSwapContext+0x7f
fffffa60`190e5ac0 fffff800`01caedab nt!KiSwapThread+0x13a
fffffa60`190e5b30 fffff800`01f1d608 nt!KeWaitForSingleObject+0x2cb
fffffa60`190e5bc0 fffff800`01cb7973 nt!NtWaitForSingleObject+0x98
fffffa60`190e5c20 00000000`75183d09 nt!KiSystemServiceCopyEnd+0x13 (TrapFrame @ fffffa60`190e5c20)
00000000`069ef118 00000000`75183b06 wow64cpu!CpupSyscallStub+0x9
00000000`069ef120 00000000`74f8ab46 wow64cpu!Thunk0ArgReloadState+0x1a
00000000`069ef190 00000000`74f8a14c wow64!RunCpuSimulation+0xa
00000000`069ef1c0 00000000`771605a8 wow64!Wow64LdrpInitialize+0x4b4
00000000`069ef720 00000000`771168de ntdll! ?? ::FNODOBFM::`string'+0x20aa1
00000000`069ef7d0 00000000`00000000 ntdll!LdrInitializeThunk+0xe

.process /p /r 0
Implicit thread is now fffffa80`1f3a3bb0
WARNING: WOW context retrieval requires
switching to the thread's process context.
Use .process /p fffffa80`1f6b2990 to switch back.
Implicit process is now fffffa80`13177c10
x86 context set
Loading Kernel Symbols
Loading User Symbols
Loading unloaded module list
Loading Wow64 Symbols
ChildEBP RetAddr
06aefc68 76921270 ntdll_772b0000!ZwWaitForSingleObject+0x15
06aefcd8 7328c639 kernel32!WaitForSingleObjectEx+0xbe
06aefd1c 7328c56f mscorwks!PEImage::LoadImage+0x1af
```

```
06aefd6c 7328c58e mscorwks!CLREvent::WaitEx+0x117
06aefd80 733770fb mscorwks!CLREvent::Wait+0x17
06aefe00 73377589 mscorwks!ThreadpoolMgr::SafeWait+0x73
06aefe64 733853f9 mscorwks!ThreadpoolMgr::WorkerThreadStart+0x11c
06aeff88 7699eccb mscorwks!Thread::intermediateThreadProc+0x49
06aeff94 7732d24d kernel32!BaseThreadInitThunk+0xe
06aeffd4 7732d45f ntdll_772b0000!__RtlUserThreadStart+0x23
06aeffec 00000000 ntdll_772b0000!_RtlUserThreadStart+0x1b
Effective machine: x64 (AMD64)

[...]
```