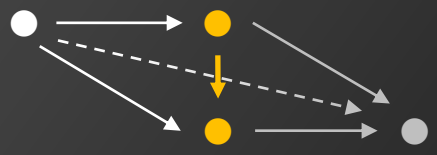
Part 3 Draft

Windows API

for Software Diagnostics

Accelerated

With Category Theory in View



Dmitry Vostokov
Software Diagnostics Services

Exports and Imports

- WinDbg (manual/scripts)
- 3rd-party WinDbg extensions (SwishDbgEx)
- DUMPBIN

API and System Calls

- API that do not require kernel services
  - `GetCurrentThreadId`
- API that require kernel services
  - `user32!CreateWindowExW → win32u!NtUserCreateWindowEx`
  - `kernel32!ReadFile → ntdll!NtReadFile`

Exercise W6

- **Goal:** Explore exports and imports using dumpbin. Check whether the selected API functions use a system call
- **ADDR Patterns:** Call Path
- \AWAPI-Dumps\Exercise-W6.pdf

Documented API

- Online documentation
- Present in headers
- Example:

Documentation

```
kernel32!SuspendThread →
        KERNELBASE!SuspendThread →
                ntdll!NtSuspendThread
```

Undocumented API

`ntdll!NtSuspendProcess`

```
0:000> x /v ntdll!*
…
pub func   00007ffc`f6346ff0    0 ntdll!NtSuspendProcess (NtSuspendProcess)
…
```

API Source Code

- Wine (GitLab) / Wine-Mirror (GitHub)

- Example:

`user32!CreateWindowExW`

https://gitlab.winehq.org/wine/wine/-/blob/master/dlls/user32/win.c
https://github.com/wine-mirror/wine/blob/master/dlls/user32/win.c

API Name Patterns

- Create/Open/Delete/Close
- Process/Thread
- Memory
- Read/Write

WinDbg Commands

```
0:000> x /v module!fpattern
```

API Namespaces

- API sets / contracts
  - Example: CreateDialogParamW
- Functions required to accomplish a particular task
  - Example: screen capture

```
gdi32!CreateCompatibleDC
gdi32!StretchBlt
gdi32!CreateDIBSection
gdi32!SelectObject

user32!ReleaseDC
user32!NtUserGetWindowDC
user32!GetWindowRect
```

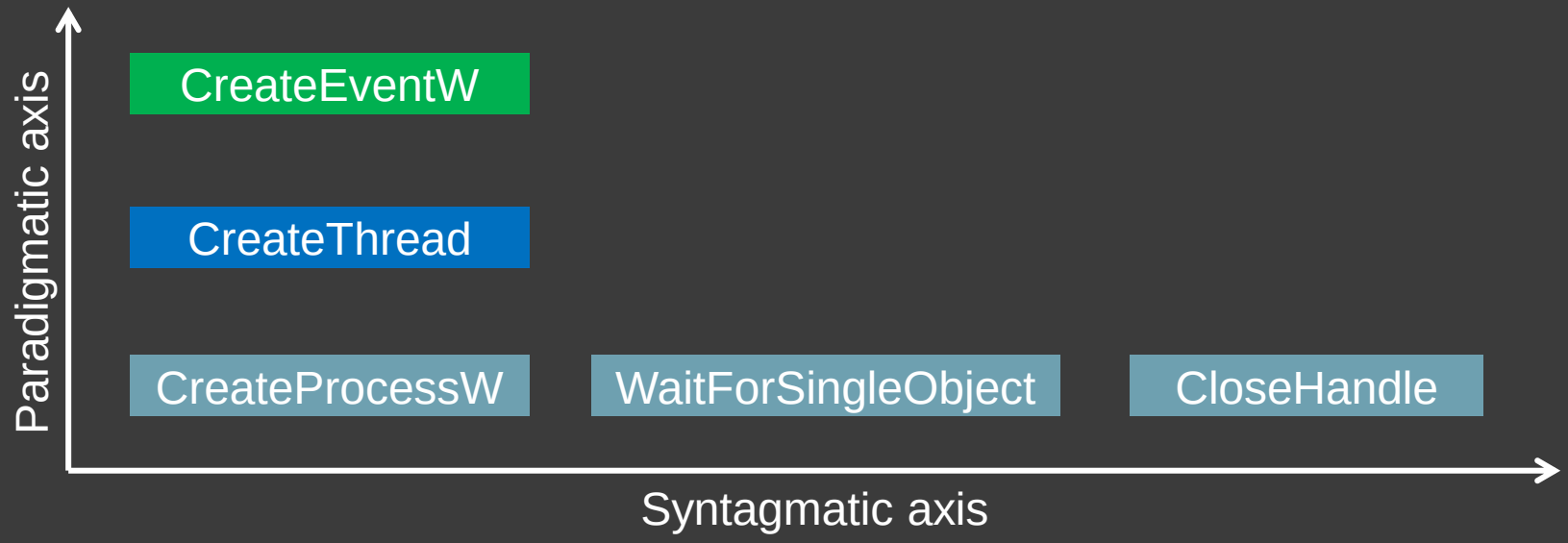
saving image

```
GdiPlus!GdiplusStartup
GdiPlus!GdipSaveImageToStream
GdiPlus!GdipGetImageEncodersSize
GdiPlus!GdipDisposeImage
GdiPlus!GdipCreateBitmapFromHBITMAP
GdiPlus!GdipGetImageEncoders

ole32!CreateStreamOnHGlobal
```

API Syntagms/Paradigms

- Syntagms / syntagmatic analysis
- Paradigms / paradigmatic analysis



Marked API

- Marked Message trace and log analysis pattern
- Points to presence or absence of activity
- Example:
  - CreateThread [-]
  - socket [+]
  - GetMessageW [-]
  - ReadConsoleW [+]

WinDbg Commands

```
0:000> x app!_imp_pattern
```

ADDR Patterns

- From Accelerated Disassembly Deconstruction Reversing
- List of pattern names
- Pattern descriptions

DebugWare Patterns

- Patterns for troubleshooting and debugging tools
- API Query

  Periodic or asynchronous query of the same set of API and logging of their input and output data.

- Example: WindowHistory

Patterns vs. Analysis Patterns

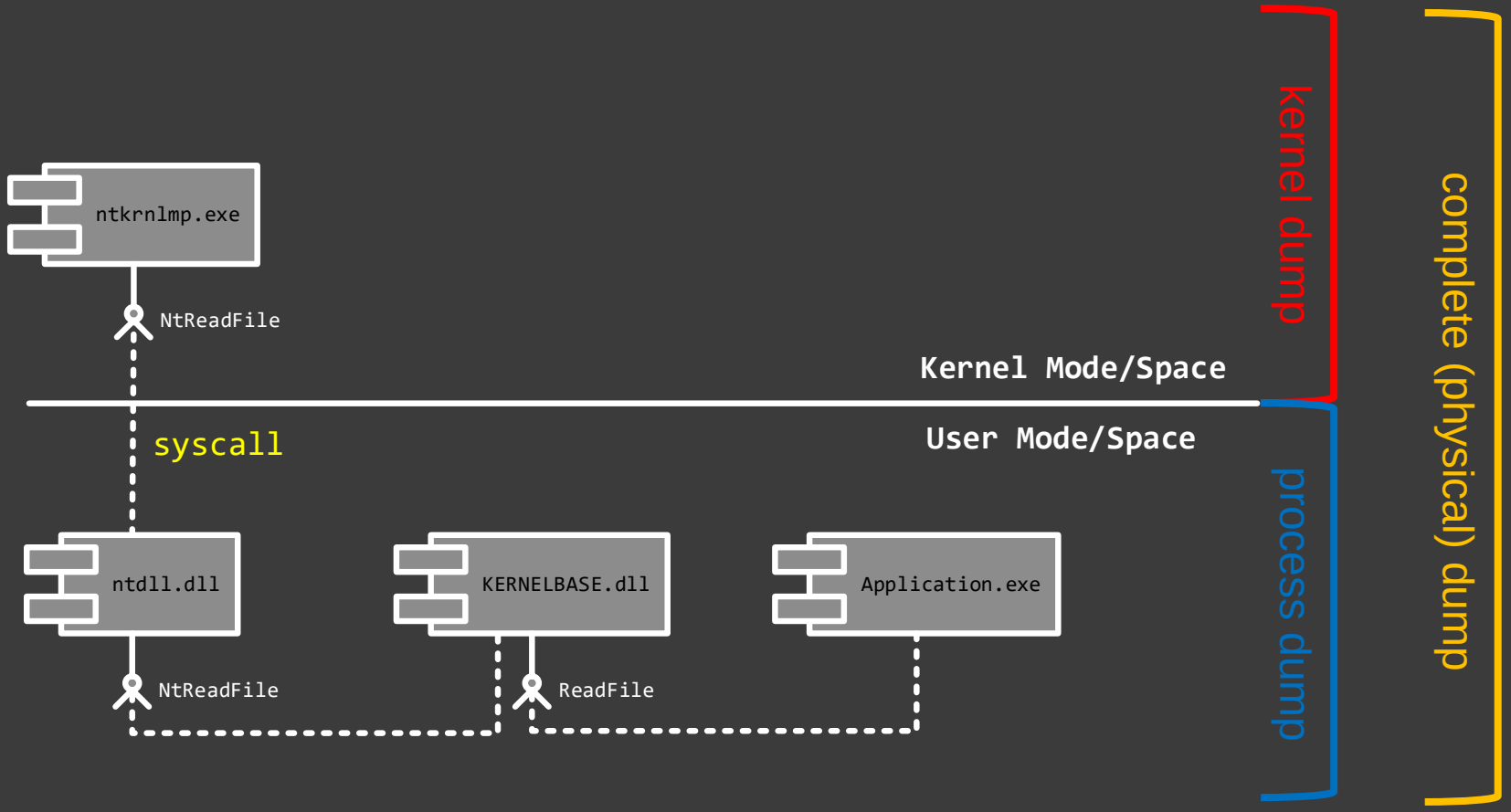
Diagnostic Pattern: a common recurrent identifiable problem together with a set of recommendations and possible solutions to apply in a specific context.

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.

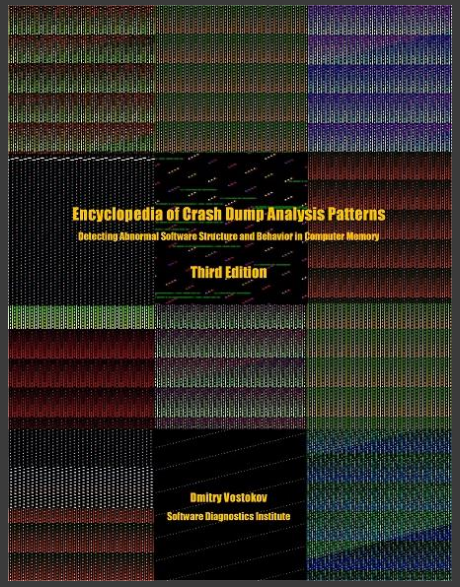
Diagnostics Pattern Language: common names of diagnostic and diagnostic analysis patterns. The same language for any operating system: Windows, Mac OS X, Linux, ...

Memory Dump Types

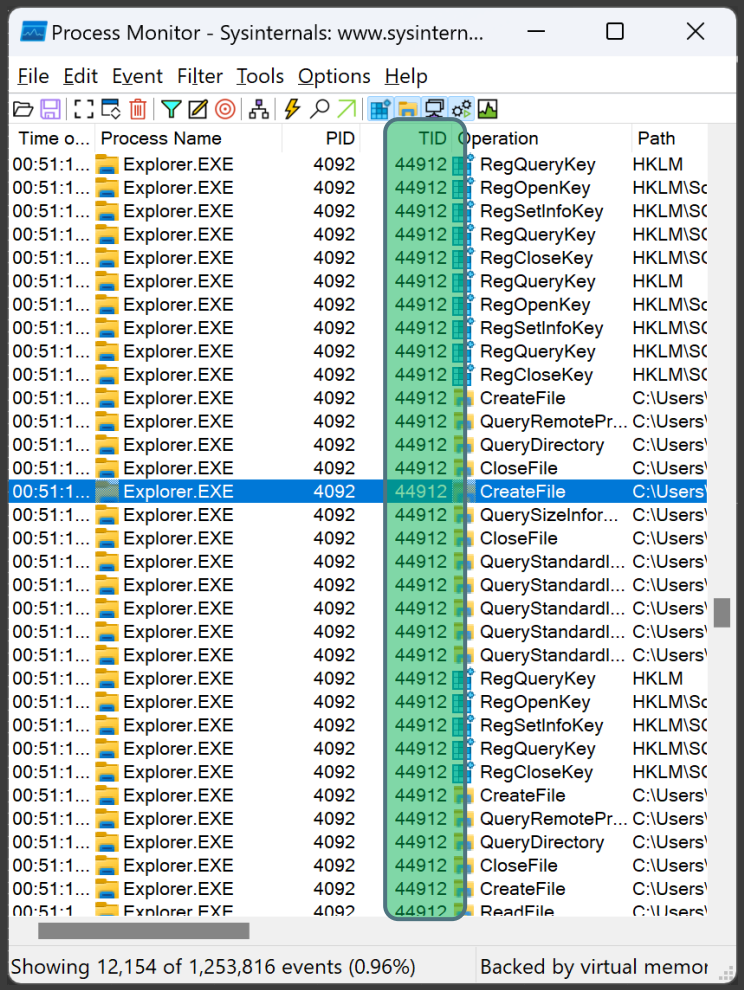


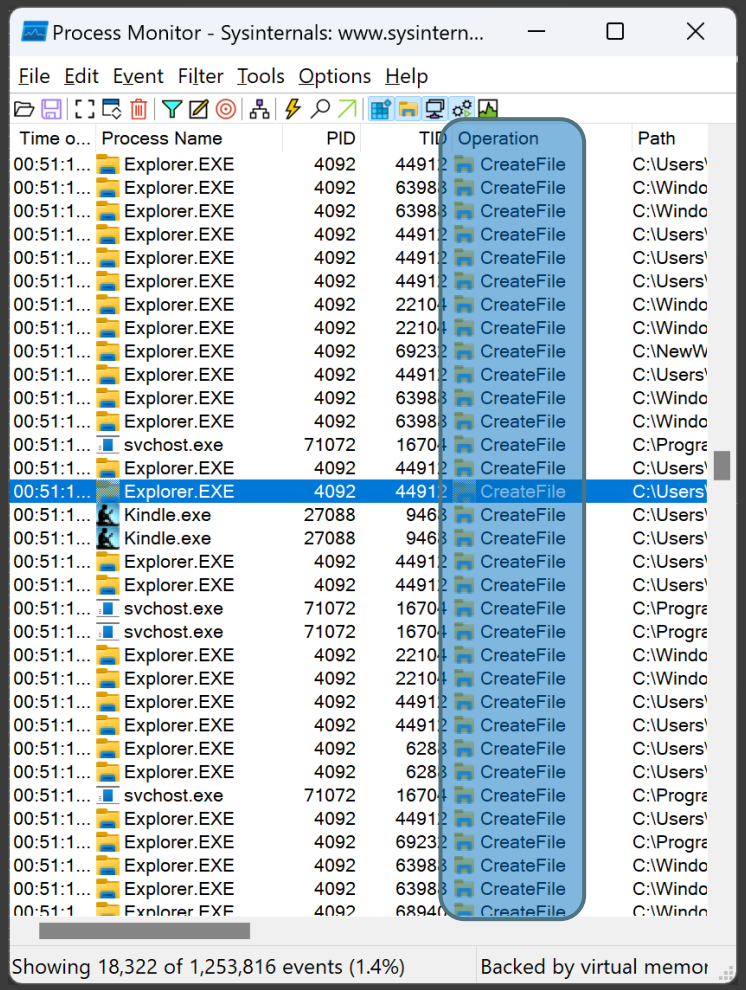
Memory Analysis Patterns

- User space
  - Process memory dumps
  - Complete memory dumps
- Function analysis patterns
  - Stack Trace Collection
  - Well-Tested Function
  - False Function Parameters
  - String Parameter
  - Small Value / Design Value
  - Virtualized Process
  - Stack Trace
  - Execution Residue
  - Hidden Parameter
  - Parameter Flow
  - Data Correlation

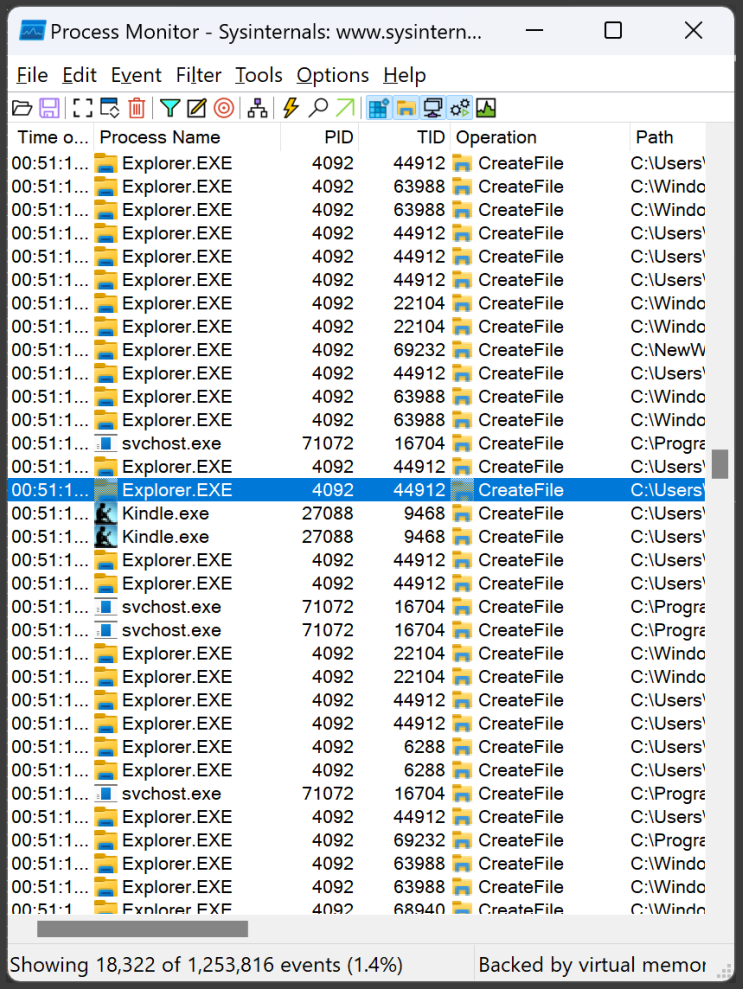


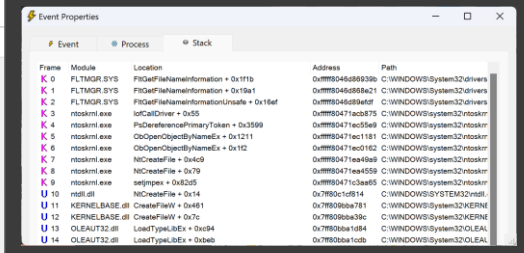
Thread and Adjoint Thread

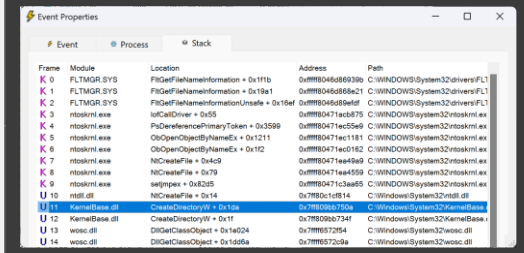


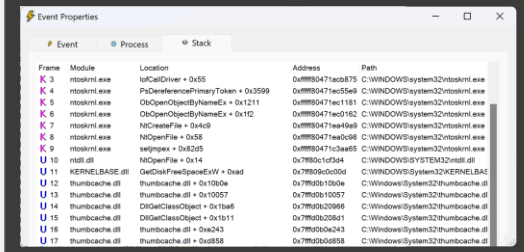


Fiber Bundle



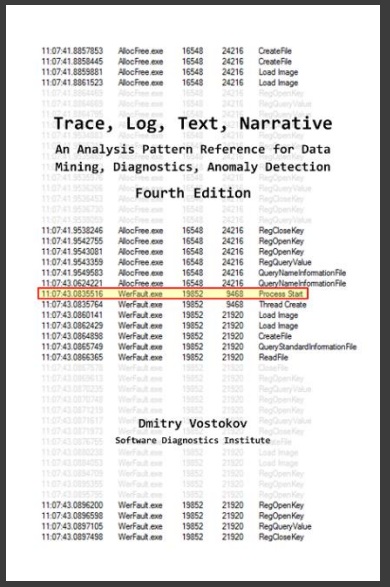






Trace and Log Analysis Patterns

- Process Monitor



- Function calls:
  - Thread of Activity
  - Fiber of Activity
  - Adjoint Thread of Activity
  - Strand of Activity
  - Discontinuity
  - Fiber Bundle
  - Weave of Activity

WOW64

- 64-bit process dumps
- 32-bit process dumps
- `wow64` `(kernel32)`
- `wow64win` `(user32, gdi32)`
- `wow64cpu`

Exercise W7

- **Goal:** Explore Windows API calls in the WOW64 context
- **Memory Analysis Patterns:** Stack Trace Collection; Virtualized Process
- \AWAPI-Dumps\Exercise-W7.pdf