Repacking the unpacker: Applying Time Travel Debugging to malware analysis

Benoît Sevens

Who am I?

- Security Consultant @ Atos Luxembourg
- Specialized in Malware Analysis and Reverse Engineering
- Had a good day when:

Agenda

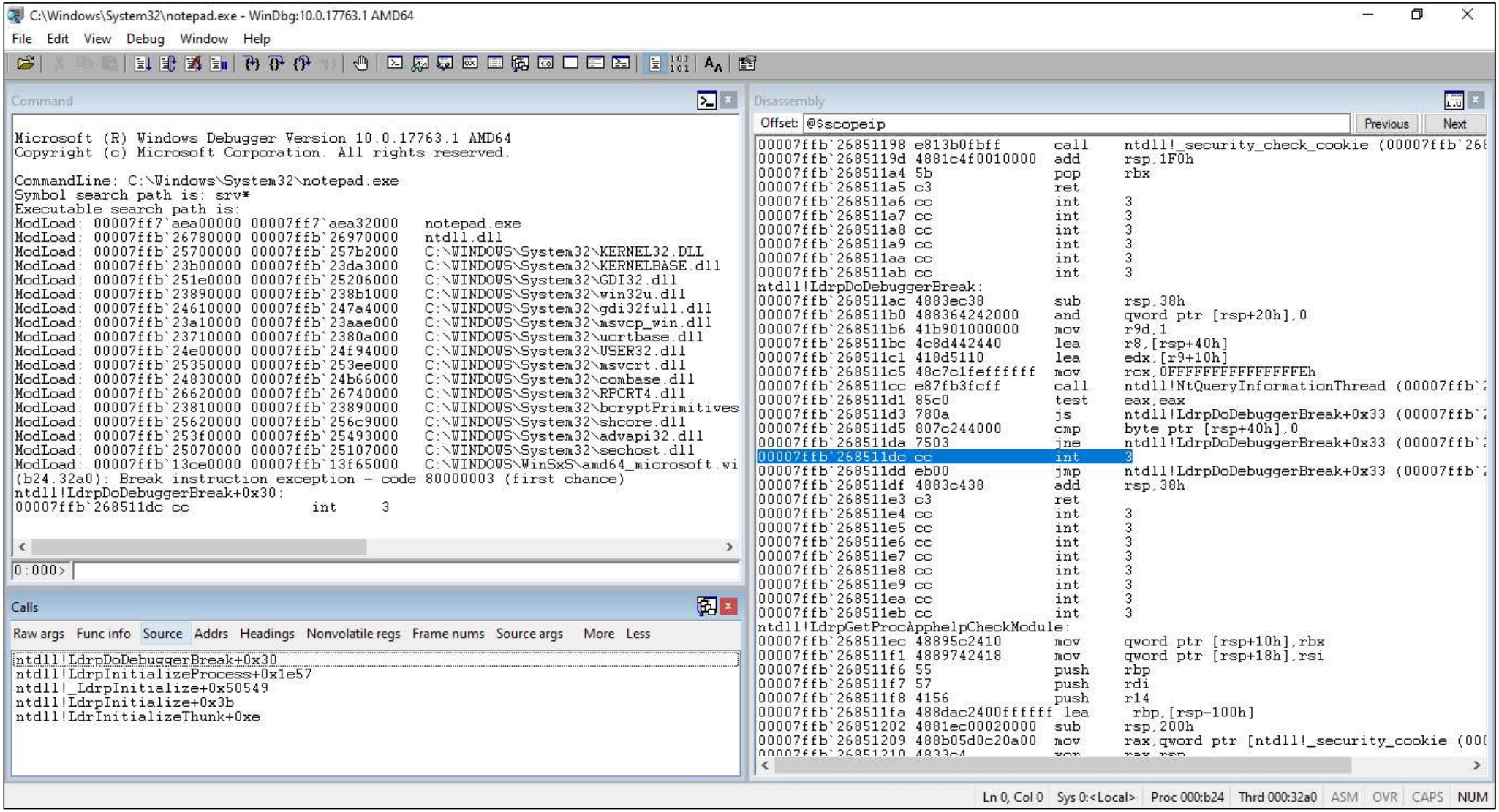
- WinDbg Preview
- Time Travel Debugging (TTD)
- Debugger data model and LINQ
- JavaScript scripting
- Some words about unpacking
- Exercises
  - Exercise 1 – `PeeAndGee.run`
  - Exercise 2 – `Reflect.run`

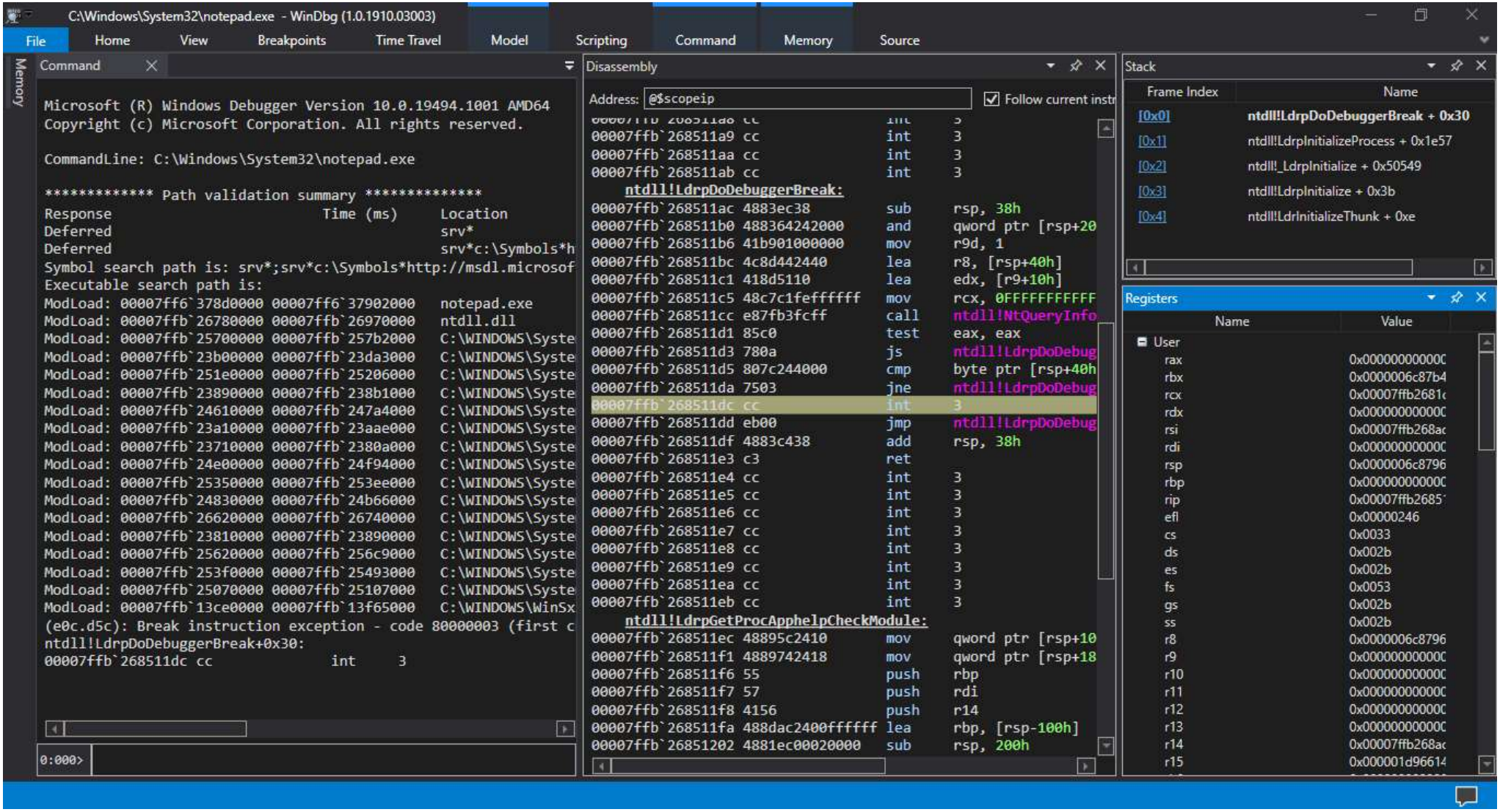
But first… the symbols!

- Connect to the internet
- Open both trace files
  - 1-PeeAndGee.run
  - 2-Reflect.run
- Download the symbols
  - `.reload /f`

WinDbg Preview

- Official Windows debugger of Microsoft
- AKA WinDbgX
- Installation and updates via Microsoft Store
- Completely portable (108 MB for WinDbgX)


C:\Windows\System32\notepad.exe - WinDbg:10.0.17763.1 AMD64
File Edit View Debug Window Help
Command
Microsoft (R) Windows Debugger Version 10.0.17763.1 AMD64
Copyright (c) Microsoft Corporation. All rights reserved.
CommandLine: C:\Windows\System32\notepad.exe
Symbol search path is: srv*
Executable search path is:
ModLoad: 00007ff7`aea00000 00007ff7`aea32000 notepad.exe
ModLoad: 00007ffb`26780000 00007ffb`26970000 ntdll.dll
ModLoad: 00007ffb`25700000 00007ffb`257b2000 C:\WINDOWS\System32\KERNEL32.DLL
ModLoad: 00007ffb`23b00000 00007ffb`23da3000 C:\WINDOWS\System32\KERNELBASE.dll
ModLoad: 00007ffb`251e0000 00007ffb`25206000 C:\WINDOWS\System32\GDI32.dll
ModLoad: 00007ffb`23890000 00007ffb`238b1000 C:\WINDOWS\System32\win32u.dll
ModLoad: 00007ffb`24610000 00007ffb`247a4000 C:\WINDOWS\System32\gdi32full.dll
ModLoad: 00007ffb`23a10000 00007ffb`23aae000 C:\WINDOWS\System32\msvcp_win.dll
ModLoad: 00007ffb`23710000 00007ffb`2380a000 C:\WINDOWS\System32\ucrtbase.dll
ModLoad: 00007ffb`24e00000 00007ffb`24f94000 C:\WINDOWS\System32\USER32.dll
ModLoad: 00007ffb`25350000 00007ffb`253ee000 C:\WINDOWS\System32\msvcrt.dll
ModLoad: 00007ffb`24830000 00007ffb`24b66000 C:\WINDOWS\System32\combase.dll
ModLoad: 00007ffb`26620000 00007ffb`26740000 C:\WINDOWS\System32\RPCRT4.dll
ModLoad: 00007ffb`23810000 00007ffb`23890000 C:\WINDOWS\System32\bcryptPrimitives
ModLoad: 00007ffb`25620000 00007ffb`256c9000 C:\WINDOWS\System32\shcore.dll
ModLoad: 00007ffb`253f0000 00007ffb`25493000 C:\WINDOWS\System32\advapi32.dll
ModLoad: 00007ffb`25070000 00007ffb`25107000 C:\WINDOWS\System32\sechost.dll
ModLoad: 00007ffb`13ce0000 00007ffb`13f65000 C:\WINDOWS\WinSxS\amd64_microsoft.wi
(b24.32a0): Break instruction exception - code 80000003 (first chance)
ntdll!LdrpDoDebuggerBreak+0x30:
00007ffb`268511dc cc int 3
0:000>
Calls
Raw args Func info Source Addrs Headings Nonvolatile regs Frame nums Source args More Less
ntdll!LdrpDoDebuggerBreak+0x30
ntdll!LdrpInitializeProcess+0x1e57
ntdll!_LdrpInitialize+0x50549
ntdll!LdrpInitialize+0x3b
ntdll!LdrInitializeThunk+0xe
Disassembly
Offset: @$scopeip
Previous
Next
00007ffb`26851198 e813b0fbff call ntdll!_security_check_cookie (00007ffb`268
00007ffb`2685119d 4881c4f0010000 add rsp,1F0h
00007ffb`268511a4 5b pop rbx
00007ffb`268511a5 c3 ret
00007ffb`268511a6 cc int 3
00007ffb`268511a7 cc int 3
00007ffb`268511a8 cc int 3
00007ffb`268511a9 cc int 3
00007ffb`268511aa cc int 3
00007ffb`268511ab cc int 3
ntdll!LdrpDoDebuggerBreak:
00007ffb`268511ac 4883ec38 sub rsp,38h
00007ffb`268511b0 488364242000 and qword ptr [rsp+20h],0
00007ffb`268511b6 41b901000000 mov r9d,1
00007ffb`268511bc 4c8d442440 lea r8,[rsp+40h]
00007ffb`268511c1 418d5110 lea edx,[r9+10h]
00007ffb`268511c5 48c7c1feffffff mov rcx,0FFFFFFFFFFFFFFFEh
00007ffb`268511cc e87fb3fcff call ntdll!NtQueryInformationThread (00007ffb`2
00007ffb`268511d1 85c0 test eax,eax
00007ffb`268511d3 780a js ntdll!LdrpDoDebuggerBreak+0x33 (00007ffb`2
00007ffb`268511d5 807c244000 cmp byte ptr [rsp+40h],0
00007ffb`268511da 7503 jne ntdll!LdrpDoDebuggerBreak+0x33 (00007ffb`2
00007ffb`268511dc cc int 3
00007ffb`268511dd eb00 jmp ntdll!LdrpDoDebuggerBreak+0x33 (00007ffb`2
00007ffb`268511df 4883c438 add rsp,38h
00007ffb`268511e3 c3 ret
00007ffb`268511e4 cc int 3
00007ffb`268511e5 cc int 3
00007ffb`268511e6 cc int 3
00007ffb`268511e7 cc int 3
00007ffb`268511e8 cc int 3
00007ffb`268511e9 cc int 3
00007ffb`268511ea cc int 3
00007ffb`268511eb cc int 3
ntdll!LdrpGetProcApphelpCheckModule:
00007ffb`268511ec 48895c2410 mov qword ptr [rsp+10h],rbx
00007ffb`268511f1 4889742418 mov qword ptr [rsp+18h],rsi
00007ffb`268511f6 55 push rbp
00007ffb`268511f7 57 push rdi
00007ffb`268511f8 4156 push r14
00007ffb`268511fa 488dac2400ffffff lea rbp,[rsp-100h]
00007ffb`26851202 4881ec00020000 sub rsp,200h
00007ffb`26851209 488b05d0c20a00 mov rax,qword ptr [ntdll!_security_cookie (00
Ln 0, Col 0
Sys 0:<Local>
Proc 000:b24
Thrd 000:32a0
ASM
OVR
CAPS
NUM


C:\Windows\System32\notepad.exe - WinDbg (1.0.1910.03003)
File
Home
View
Breakpoints
Time Travel
Model
Scripting
Command
Memory
Source
Memory
Command
Microsoft (R) Windows Debugger Version 10.0.19494.1001 AMD64
Copyright (c) Microsoft Corporation. All rights reserved.
CommandLine: C:\Windows\System32\notepad.exe
************* Path validation summary **************
Response Time (ms) Location
Deferred srv*
Deferred srv*c:\Symbols*h
Symbol search path is: srv*;srv*c:\Symbols*http://msdl.microsof
Executable search path is:
ModLoad: 00007ff6`378d0000 00007ff6`37902000 notepad.exe
ModLoad: 00007ffb`26780000 00007ffb`26970000 ntdll.dll
ModLoad: 00007ffb`25700000 00007ffb`257b2000 C:\WINDOWS\Syste
ModLoad: 00007ffb`23b00000 00007ffb`23da3000 C:\WINDOWS\Syste
ModLoad: 00007ffb`251e0000 00007ffb`25206000 C:\WINDOWS\Syste
ModLoad: 00007ffb`23890000 00007ffb`238b1000 C:\WINDOWS\Syste
ModLoad: 00007ffb`24610000 00007ffb`247a4000 C:\WINDOWS\Syste
ModLoad: 00007ffb`23a10000 00007ffb`23aae000 C:\WINDOWS\Syste
ModLoad: 00007ffb`23710000 00007ffb`2380a000 C:\WINDOWS\Syste
ModLoad: 00007ffb`24e00000 00007ffb`24f94000 C:\WINDOWS\Syste
ModLoad: 00007ffb`25350000 00007ffb`253ee000 C:\WINDOWS\Syste
ModLoad: 00007ffb`24830000 00007ffb`24b66000 C:\WINDOWS\Syste
ModLoad: 00007ffb`26620000 00007ffb`26740000 C:\WINDOWS\Syste
ModLoad: 00007ffb`23810000 00007ffb`23890000 C:\WINDOWS\Syste
ModLoad: 00007ffb`25620000 00007ffb`256c9000 C:\WINDOWS\Syste
ModLoad: 00007ffb`253f0000 00007ffb`25493000 C:\WINDOWS\Syste
ModLoad: 00007ffb`25070000 00007ffb`25107000 C:\WINDOWS\Syste
ModLoad: 00007ffb`13ce0000 00007ffb`13f65000 C:\WINDOWS\WinSx
(e0c.d5c): Break instruction exception - code 80000003 (first c
ntdll!LdrpDoDebuggerBreak+0x30:
00007ffb`268511dc cc int 3
0:000>
Disassembly
Address: @$scopeip
Follow current instr
00007ffb`268511a9 cc int 3
00007ffb`268511aa cc int 3
00007ffb`268511ab cc int 3
ntdll!LdrpDoDebuggerBreak:
00007ffb`268511ac 4883ec38 sub rsp, 38h
00007ffb`268511b0 488364242000 and qword ptr [rsp+20
00007ffb`268511b6 41b901000000 mov r9d, 1
00007ffb`268511bc 4c8d442440 lea r8, [rsp+40h]
00007ffb`268511c1 418d5110 lea edx, [r9+10h]
00007ffb`268511c5 48c7c1feffffff mov rcx, 0FFFFFFFFFF
00007ffb`268511cc e87fb3fcff call ntdll!NtQueryInfo
00007ffb`268511d1 85c0 test eax, eax
00007ffb`268511d3 780a js ntdll!LdrpDoDebug
00007ffb`268511d5 807c244000 cmp byte ptr [rsp+40h
00007ffb`268511da 7503 jne ntdll!LdrpDoDebug
00007ffb`268511dc cc int 3
00007ffb`268511dd eb00 jmp ntdll!LdrpDoDebug
00007ffb`268511df 4883c438 add rsp, 38h
00007ffb`268511e3 c3 ret
00007ffb`268511e4 cc int 3
00007ffb`268511e5 cc int 3
00007ffb`268511e6 cc int 3
00007ffb`268511e7 cc int 3
00007ffb`268511e8 cc int 3
00007ffb`268511e9 cc int 3
00007ffb`268511ea cc int 3
00007ffb`268511eb cc int 3
ntdll!LdrpGetProcApphelpCheckModule:
00007ffb`268511ec 48895c2410 mov qword ptr [rsp+10
00007ffb`268511f1 4889742418 mov qword ptr [rsp+18
00007ffb`268511f6 55 push rbp
00007ffb`268511f7 57 push rdi
00007ffb`268511f8 4156 push r14
00007ffb`268511fa 488dac2400ffffff lea rbp, [rsp-100h]
00007ffb`26851202 4881ec00020000 sub rsp, 200h
Stack
Frame Index
Name
[0x0] ntdll!LdrpDoDebuggerBreak + 0x30
[0x1] ntdll!LdrpInitializeProcess + 0x1e57
[0x2] ntdll!_LdrpInitialize + 0x50549
[0x3] ntdll!LdrpInitialize + 0x3b
[0x4] ntdll!LdrInitializeThunk + 0xe
Registers
Name
Value
User
rax
rbx 0x0000006c87b4
rcx 0x00007ffb2681
rdx
rsi 0x00007ffb268a
rdi
rsp 0x0000006c8796
rbp
rip 0x00007ffb2685
efl 0x00000246
cs 0x0033
ds 0x002b
es 0x002b
fs 0x0053
gs 0x002b
ss 0x002b
r8 0x0000006c8796
r9
r10
r11
r12
r13
r14 0x00007ffb268a
r15 0x000001d96614

Launching WinDbgX

- Installed versison
  - From PATH: `C:\> WinDbgX`
  - From Start Menu
- Portable version:
  - `C:\WinDbgX> DbgX.Shell.exe`
- Double-click file of associate file type

Time Travel Debugging (TTD)

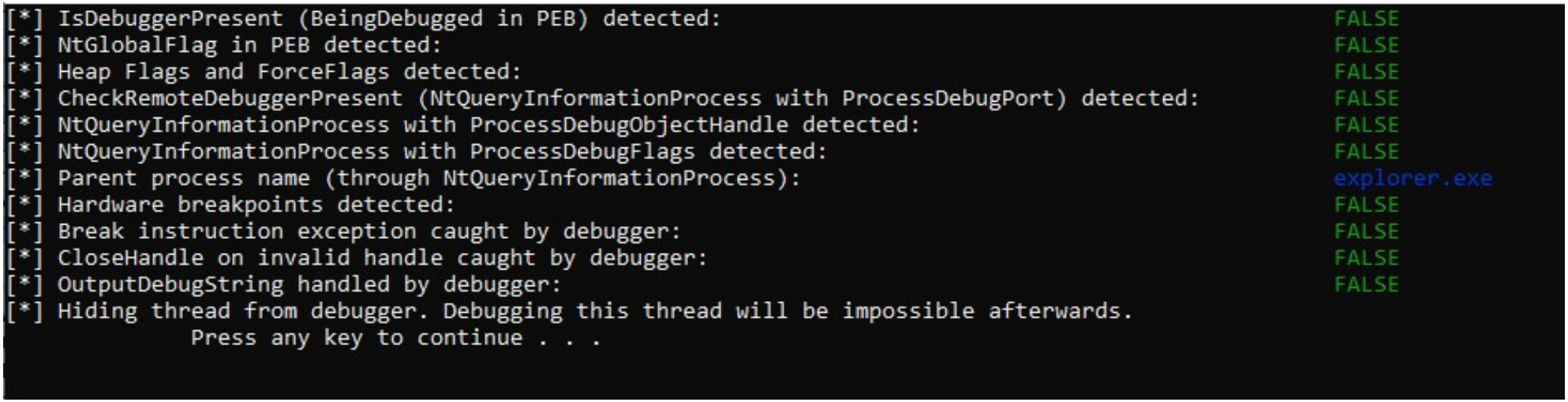
- Introduced in WinDbg Preview
- Records execution as a movie
- Generates a trace file that can be replayed forwards and backwards
- Can also be used as a sort of database that can be queried
  - E.g.: Give me all calls to `CreateProcess()`
  - E.g.: Give me all memory writes to address X
- Different debugging experience
- First tool to support this for native code on Windows

Use cases and advantages

- Deterministic replay of execution trace
  - Work with colleagues on exact same data
  - No more “Damn, I missed it”
- No risk for infection when working on the trace
- Not really debugging, so no debugging detection

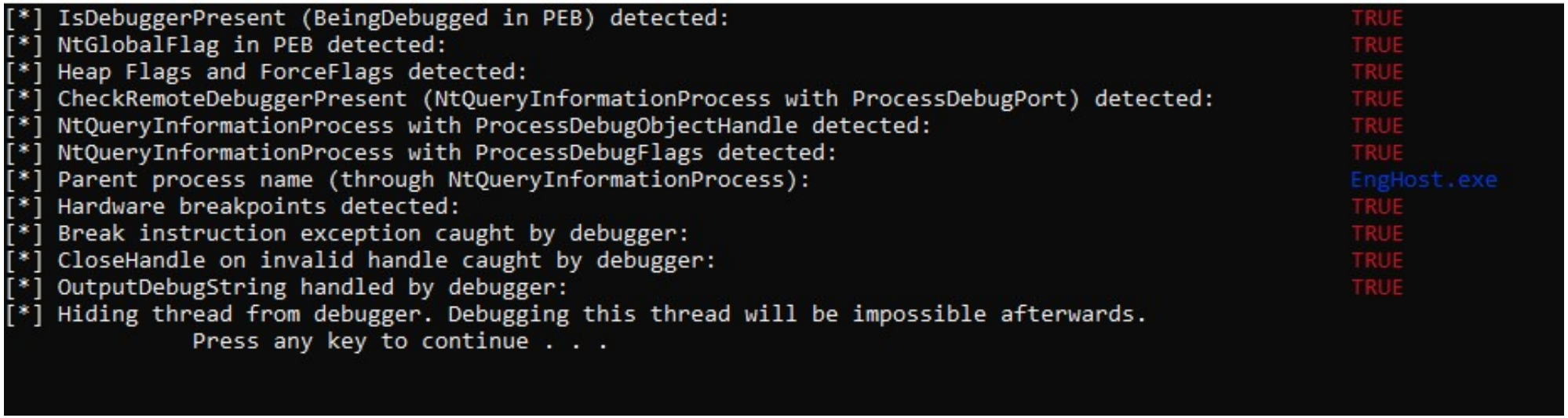
Debugger detection

No debugger attached:



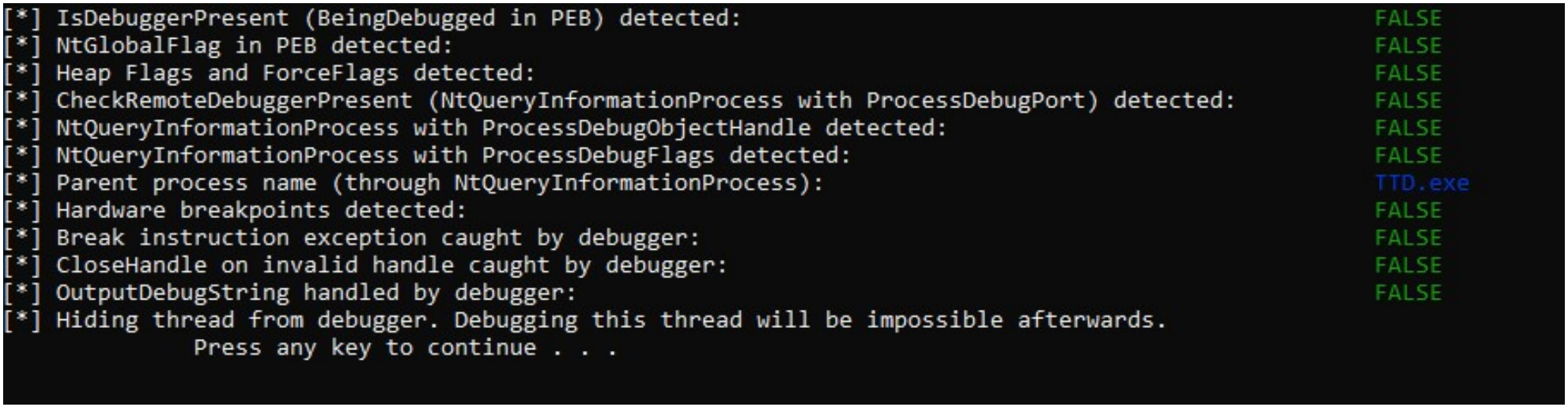
Debugger detection

Debugger attached:



Debugger detection

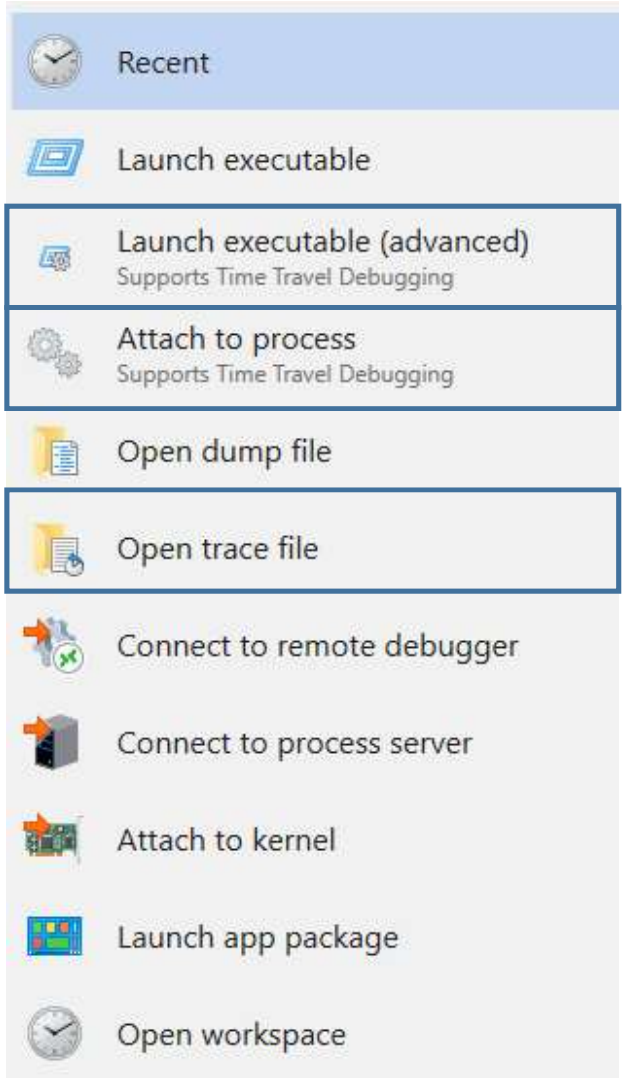
Time Travel “Debugged”:



Limitations

- Must run WinDbgX as administrator
- Only works on user mode executables
- Anti-virus might hinder recording (likely disabled in sandbox)
- No patching of memory possible
- Can only record a process and its children
- No debugging of PPL
- Based on emulation
  - Possible “derailment”

Supported modes



Start tracing

- Method 1: GUI
- Method 2: command-line
  - `ttd –out sample.run (-children) sample.exe`

Files

- Logging: `.OUT`
- Trace files: `.RUN`
- Index files: `.IDX`
  - Automatically created when opening trace file
- `.RUN` file is the only file you need to share
- Be sure to have enough disk space when recording
- Each position in a trace file has a “coordinate”

New TTD commands

- `g-, p-, t-` **Go, step, trace backwards**
- `!tt` **Travel to position**
- `!index` Index time travel trace
- `!positions` Display all active threads with their positions

Debugger data model and LINQ

- Accessible via:
  - **GUI:** Dedicated “model” window
  - **Command line:** `dx` command
  - **JavaScript:** `host.namespace.Debugger` namespace
- Handy “shortcut” variables: `$curprocess`, `$curthread` and `$cursession`
- New Debugger Object Model objects introducted by TTD
  - `dx @$curthread.TTD`
  - `dx @$curprocess.TTD`
  - `dx @$cursession.TTD`
    - **`dx @$cursession.TTD.Calls`**
    - **`dx @$cursession.TTD.Memory`**

Debugger data model and LINQ

- Can be combined with (C#) Language Integrated Query (LINQ) syntax
- Some examples of methods:
  - `dx @$cursession.TTD.Calls("kernel32!VirtualAlloc*").Count()`
  - `dx @$cursession.TTD.Calls("kernel32!VirtualAlloc*").Select(c => c.Parameters)`
  - `dx @$cursession.TTD.Calls("kernel32!VirtualAlloc*").Where(c => c.Parameters[3] == 0x40)`
  - `dx @$cursession.TTD.Calls("kernel32!VirtualAlloc*").OrderBy(c => c.TimeStart)`
- `dx -g` formats iterable output as a pretty table

JavaScript scripting

- WinDbg comes with a JavaScript engine (`ChakraCore.dll`)
- Debugger data model accessible from `host.namespace.Debugger` namespace

Some words about unpackers

- Typically, unpackers allocate memory for the unpacked code
- This memory has to be **writeable** at some moment, and **executable** at another moment
- Multiple API calls are possible, some which can control the memory protections:
  - **`VirtualAlloc`**
  - **`VirtualAllocEx`**
  - **`NtAllocateVirtualMemory`**
  - `GlobalAlloc, HeapAlloc, LocalAlloc`
  - `malloc, calloc`
  - `RtlAllocateHeap`
  - `CoTaskMemAlloc`
- In order to change memory protections:
  - `VirtualProtect`
  - `VirtualProtectEx`
  - `NtProtectVirtualMemory`

Commands we will need

- `db` Display memory
- `dx` Display debugger object model expression
- `t(-)` Trace (back)
- `.writemem` Write memory to file
- `!dh` Displays header of image
- `!tt` Time travel

Time for exercises!

- Two exercises:
  - 1-PeeAndGee.run
  - 2-Reflect.run
- See exercises workbook

Questions

```
0:000> .hh ?
```

References and resources

- Blog posts
  - hugsy - Some Time Travel Musings
  - Axel "0vercl0k" Souchet - Debugger data model, Javascript & x64 exception handling
- Documentation
  - Microsoft - Debugging Tools for Windows
- Workshops
  - hugsy, 0vercl0k - Modern Debugging with WinDbg Preview
- Talks
  - Channel 9 - Defrag Tools
  - J. McNellis, J. Mola, K. Sykes - Time Travel Debugging: Root Causing Bugs in Commercial Scale Software
  - Daniel Pearson - Windows Debugging and Troubleshooting
  - James McNellis - Time Travel Debugging
  - What's new in WinDbg Preview - Andy Luhrs