

Pranešėjas: Kęstas Gudinavičius

# DERINIMAS SU WINDBG

# Kas yra WinDBG?

- Galingas grafinis derintuvas skirtas Windows tvarkyklių, tarnybų ir paprastų aplikacijų derinimui
- Debugging Tools for Windows paketo dalis

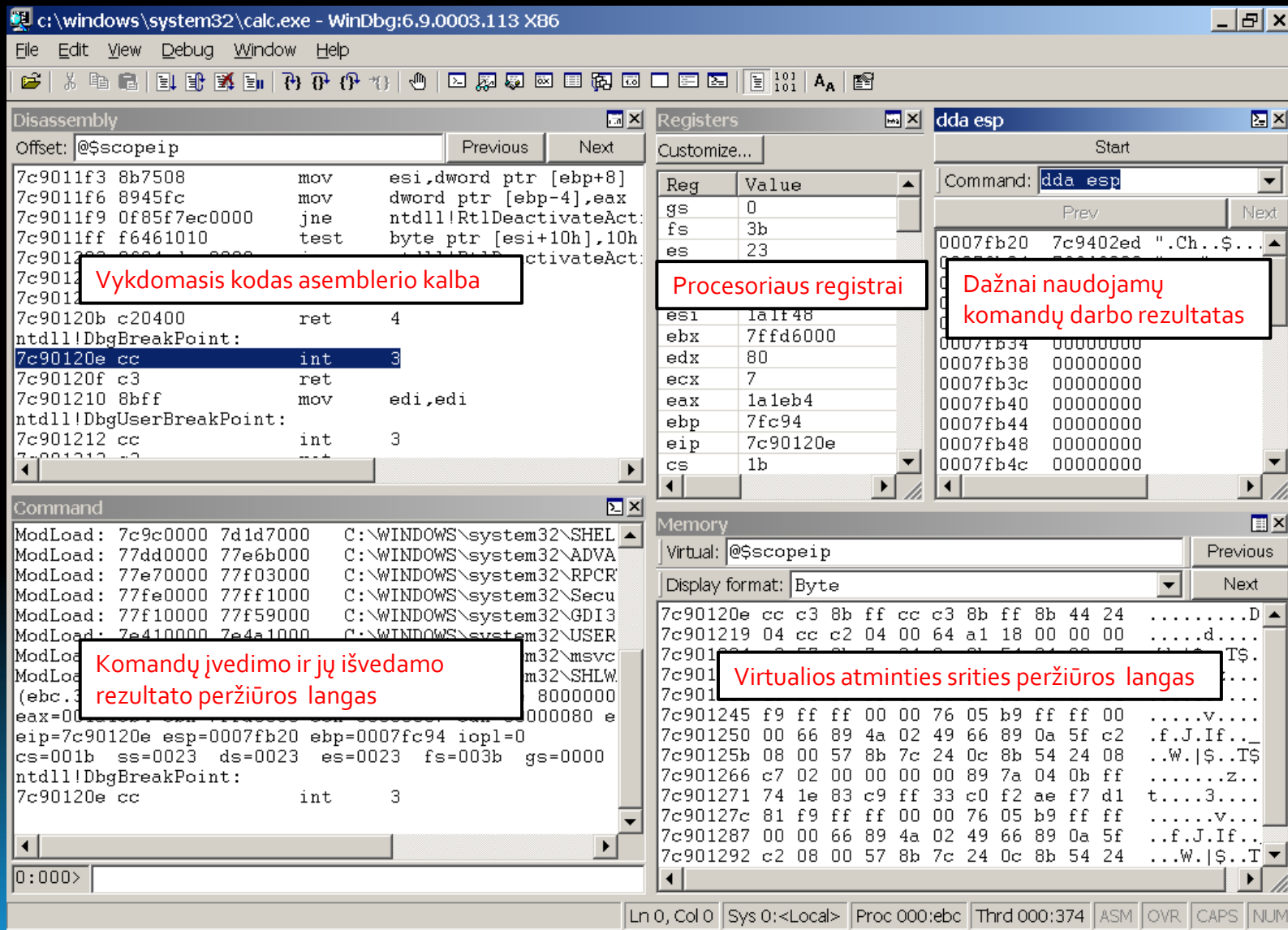
# Ką sugeba WinDBG?

- User-mode ir kernel-mode derinimas
- Vietinis ir nutolęs derinimas
- Automatinė derinimo simbolių įkrova
  - Koreliacija su išeities tekstais
- Atminties dumpų analizė
- Plėtinių palaikymas
- Scenarijų rašymas (scripting)
- 32 ir 64 bitų palaikymas

# Derinimo seanso pradžia

- WinDBG darbo aplinkos nustatymas
- Derinimo simbolių nustatymas
- Proceso paleidimas po derintuvu
- Prisijungimas prie proceso
- Post-mortem derinimas
- Atminties dumpo analizė

# WinDBG darbo aplinkos nustatymas



The screenshot shows the WinDBG interface with several key components and annotations:

- Disassembly:** Shows assembly code for `calc.exe`. The instruction `7c90120e cc int 3` is highlighted. A red box with the text "Vykdomas kodas assemblerio kalba" (Executable code in assembler language) points to this instruction.
- Registers:** Lists the current values of CPU registers. A red box with the text "Procesoriai registrai" (Processor registers) points to this window.
- Command Window:** Shows the command `dda esp` and its output, including memory addresses and values. A red box with the text "Dažnai naudojamų komandų darbo rezultatas" (Result of frequently used commands) points to the output.
- Command Window (bottom):** Shows the command prompt `0:000>`. A red box with the text "Komandų įvedimo ir jų išvedamo rezultato peržiūros langas" (Command input and output result viewing window) points to this area.
- Memory Window:** Shows the memory dump for the current instruction. A red box with the text "Virtualios atminties srities peržiūros langas" (Memory window) points to this area.

# Derinimo simbolių nustatymas

- Linkerio sugeneruoti .PDB failai
- Derinimo informacija įtraukta į patį vykdomąjį failą
- Simbolių serveriai
  - File>Symbol Path
  - Windows bibliotekoms (DLL):  
SRV\*C:\websymbols\*http://msdl.microsoft.com/  
download/symbols

# Demo 1



# Proceso paleidimas po derintuvu

- Visiška proceso kontrolė
  - ▣ Nėra vykdomas joks kodas
- Debug heap
  - ▣ Heapo bloko vientisumo užtikrinimas
  - ▣ Gali būti išjungtas nustatius aplinkos kintamąjį `_NO_DEBUG_HEAP` reikšmę į 1
- File>Open Executable

# Prisijungimas prie proceso

- Specifinių kodo dalių analizė praleidžiant įkrovos kodą
- File>Attach to a Process

# Post-mortem derinimas

- User-mode išimtinių situacijų valdymas
- HKLM\Software\Microsoft\Windows NT\CurrentVersion\AeDebug
  - ▣ Auto
  - ▣ Debugger
- Dr. Watson – saugo atminties dumpus ir/arba šiunčia juos Microsoftui
- WinDBG nustatymas: **windbg.exe -I**

# Demo 2



# Atminties dumpo analizė

- Standartiškai Windows sistemos sukuria atminties dumpą įvykus BSoD
- Windows XP atveju 64 KB .dmp failą
  - C:\WINDOWS\minidump\
- File>Open Crash Dump
- !analyze -v

# Demo 3



# WinDBG komandų rūšys

- Standartinės komandos taikomos derinimo sesijai
  - g, kb, bp, lm
- Meta komandos taikomas pačiam derintuvui
  - .load, .tlist, .attach
- Plėtinių komandos
  - !analyze, !heap, !peb

# WinDBG sintaksė

- Skaičiai ir operatoriai
  - 0x10 arba 10h
  - 0x0018fd10+100h, ebp-8
  - poi(ebp-4)
- Adresų ruožai
  - 0x00001000 0x00001007
  - 0x00001000 L8
  - L?FFFFFFFF

# WinDBG pagrindinių komandų grupės

- Modulių peržiūra
- Simbolių nagrinėjimas
- Duomenų tipų nagrinėjimas
- Kodo vykdymas
- Breakpointai
- Procesoriaus registrai
- Atminties operacijos
- Išimtinės situacijos
- Stekas
- Heapas

# Moduļiņ peržiūra

- Im
  - Im f
  - Im m \*kernel\*
  - Im vm kernel32
- **!dh ImageBaseAddr** – peržiūrēti PE failo informacijā

# Simbolių nagrinėjimas

- x Module!Symbol
  - x \*!\*
    - x kernel32!\*
      - x user32!\*MessageBox\*
- **.reload /f** - apeinamas „lazy symbol loading“

# Duomenų tipų nagrinėjimas

- dt [mod!]Name [addr]
  - dt ntdll!\*PEB\*
  - dt ntdll!\_PEB
  - dt ntdll!\_PEB 0x7ffdd000

# Demo 4



# Kodo vykdymas

- **g** – pradėti kodo vykdymą
  - **gu** – įvykdyti esamą funkciją
- **p** – vykdyti vieną instrukciją, funkcijų kvietimą traktuoti kaip vieną žingsnį
  - **pt** – vykdyti iki funkcijos pabaigos (ret)
- **t** – vykdyti vieną instrukciją
  - **tt** – vykdyti iki artimiausios ret instrukcijos

# Breakpointai

- **bp** – nustatyti programinį bp
  - bp addr
  - bp USER32!MessageBoxExA+0x16
  - bp module!function "j ecx = 0x01 ' '; 'gc'"
- **ba** – nustatyti hardvarinį bp
  - ba [r|w|e] [size] addr
- **bl** – peržiūrėti bp
- **bc** – trinti bp
  - bc \*
  - bc 3

# Demo 5



# Procesoriaus registrai

- **r** – peržiūrėti/nustatyti registų reikšmes
  - `r eax, ebx`
  - `r eax=10`
- **View>Registers**

# Atminties operacijos

- **d\*** [/c #] **addr** – parodyti atminties sritį
  - dd – dword formatu
  - da – ascii eilutė iki null baido
- **e\*** – redaguoti atmintį
- **dd\*** – dereferencina atmintį
- **!address** – atminties srities informacija
- **s range patter** – paieška atmintyje
- **s -a 0x00000000 L?ffffffff "ABC"**
- **s 0x0018f926 L20 FF E4**

# Atminties operacijos

- **u** – atminties srities translacija į assemblerio instrukcijas (disasemblinimas)
  - u eip
  - ub eip
- u USER32!MessageBoxExA L20
- **a** – asemblio instrukcijų translacija į opkodus
- View>Dissassembly

A vertical bar on the left side of the slide, consisting of several colored segments: a white segment at the top, followed by a dark blue segment, a light blue segment, and a larger blue segment at the bottom.

# Demo 6

# Išimtinės situacijos

- **!exchain** – parodo išimtinių situacijų apdorojimo funkcijų grandinę
- **gH** – pažymėti išimtinę situacija kaip suvaldytą
- **gN** – perduoti valdymą išimtinių situacijų apdorojimo funkcijai

# Demo 7



# Stekas

- **k** – parodo call steką
  - **kb** – call stekas su pirmais 3 funkcijos parametrais
- View>Call Stack

# Heapas

- **!heap** – parodo proceso heapus
  - ▣ !heap -s
- **!heap -a addr** – informacija apie heapą
- **!heap -i addr** – informacija apie heapo bloką
- **!heap -x -v addr** – heapo bloko, kuriam priklauso adresas paieška
- **!heap -p** – operacijos su pageheapu

# Demo 8



# WinDBG plėtiniai

- C:\Program Files\Debugging Tools for Windows (x86)\winext\
  - .load DLLName
    - ▣ !extension [arguments]
- ![module.]extension [arguments]

# WinDBG plėtiniai: !exploitable

- Tariausi „bang exploitable“
- .load msec; !exploitable
- Automatinė crashų analizė siekiant įvertinti saugumo riziką
  - ▣ Exploitable
  - ▣ Probably Exploitable
  - ▣ Probably Not Exploitable
  - ▣ Unknown
- Būtina papildoma analizė

# WinDBG plėtiniai: narly

- Parodo užkrautų modulių informaciją
  - ▣ /SafeSEH
  - ▣ /GS
  - ▣ DEP
  - ▣ ASLR
- .load narly; !nmod
- Ateityje žadama daugiau funkcionalumo

# WinDBG plėtiniai: pykd

- Python modulis ir kartu WinDBG plėtinys
- Derinimo automatizacija Python kalba
- `.load pykd.pyd; !py c:\test.py`
  - ▣ `%PYTHONPATH %`
- Jeigu netenkina standartinė scenarijų rašymo galimybė

# Demo 9





# Klausimai?