CodeMachine

Security Research, Development & Training

Kernel Debugger Command Cheat Sheet

Command	Description
version	Displays target computer version
.sympath | .sympath+ <Path>	Displays debugger symbol path or appends to the current symbol path
!lmi <Module>	Displays PDB files information for a module
!chksym <Module>	Checks if the PDB file matches the module in memory
bl	Lists all the current breakpoints
bp <Symbol> | <Address>	Sets an execution breakpoint on an address
bu <Symbol>	Sets a deferred (unresolved) execution breakpoint on a symbol
ba <e|r|w> <1|2|4|8> <Address>	Sets a memory access (Read, Write, Execute) breakpoint
be | bd | bc <BreakPointId>	Enables, disables or clears a previously set breakpoint
g | p | t	Continues execution | Steps over functions | Traces into functions
db/dw/dd/dq/dp <Address>	Displays Bytes/Word/DWord/QuadWord/Pointer contents at the given address
dps <Address>	Displays pointers at the given address and resolves them to symbols
dt <Pointer> <Type>	Typecasts a pointer to the data/structure type and displays its contents
kvn	Displays the verbose call stack along with frame numbers
lmm <ModuleWildcard>	Displays the list of modules filtered by the ModuleWildcard (i.e. '*' and '?').
ln <Address>	Displays the symbols in the vicinity of the given address
ub <InstructionAddress>	Disassembles backwards from the specified address.
uf <FunctionAddress>	Performs a recursive descent disassembly of the given function
x <Symbol>	Displays the virtual address at which the given symbol is present
r | r <Register>	Displays all Registers of current CPU | Displays the given register contents
?? <C/C++ Expression>	Evaluates the given C/C++ expression (e.g. ?? this, ?? sizeof(variable))
? <MASM Expression>	Evaluates the given MASM expression (e.g. ? @$ptrsize)
!devobj <DeviceObjectPointer>	Displays the DEVICE_OBJECT structure, specified by the pointer
!devstack <DeviceObjectPointer>	Displays the stack of layered DEVICE_OBJECT structures.
!drvobj <DriverObjectName>	Displays the DRIVER_OBJECT structure at the given address.
!dh <ModuleName>	Displays the PE header information of a module loaded in memory
!irql	Displays the interrupt request level of the current CPU
!idt	Displays the interrupt descriptor table of the current CPU
!handle 0 3 <Process> File	Displays details of all the handles to FILE_OBJECTs in the process
!irp <PointerToIrp>	Displays the IRP and the IO_STACK_LOCATION structures
!object <Name> | <Pointer>	Displays an object by name (e.g. \Global??\COM1) or pointer
!pcr	Displays the kernel processor control region (KPCR) of the current CPU
!pool <PoolBlockPointer>	Displays the pool header, tag and size for the pool block
!process <Pointer> | <Id> <Flags>	Displays information about the process by EPROCESS pointer or Process ID.
.process /r /P <Pointer>	Switches the debuggers user mode process context to the specified process
!pte <VA>	Displays the page table entries for the given virtual address
!thread	Displays information about the current thread
.thread <Pointer>	Switches the debuggers register context to that of the specified thread
!vm 1	Displays overview of virtual memory state of the system
!vad <VadRoot>	Displays the virtual address descriptor (VAD) tree of the process