

Counter-measures (anti-debug, ...) cheat sheet

name	flags	description
BeingDebugged	D.....	<i>PEB.BeingDebugged</i> db [fs:[30] + 2] == 1
IsDbgPresent	D...A	<i>BeingDebugged</i> check, via <i>IsDebuggerPresent</i>
NtGlobalFlag	D.....	<i>PEB.NtGlobalFlag</i> dd [fs:[30] + 68] has 70 ¹ set
HeapFlags	D.....	<i>Heap.Flags</i> dd [[fs:[30] + 18] + C] == 2
ForceFlags	D.....	<i>Heap.ForceFlags</i> dd [[fs:[30] + 18] + 10] is not null
msvcrt!trigo	D...A	<i>msvcrt!_CIasin(invalid)</i> → a1 = NtGlobalFlag ? a8 : 98
deletefiber	D...A	<i>DeleteFiber(invalid)</i> → LastError = ForceFlags ? 80000003 : 57
gs	.SE...	GS is reset, on thread switch
pop ss	.S....	debuggers can't step right after pop SS → TF set in EFlags, 100 via pushf
smsw	.SE...	<i>operand</i> = just after FPU ? 80010031 : 8001003b
int 2c/2e	.SEI..	slides over next instruction + sets EDX to next EIP, but incorrect if stepped
int 2d	D...X	triggers <i>BREAKPOINT</i> exception if not under a debugger
InvalidHandle	D...XA	<i>CloseHandle(invalid)</i> → INVALID HANDLE exception if debugger is present
ChkRemoteDbg	D...A	<i>CheckRemoteDebuggerPresent(GetCurrentProcess()², &result) = 1 : 0</i>
NtQueryInfo	D...A	<i>NtQueryInformationProcess(-1, ProcessDebugPort = 7, var,</i> → [var] = present? - 1 : 0
HideThread	D...A	<i>NtSetInformationThread(-2, ThreadhideFromDebugger= 11, -1, 0)</i> → not responding
csr	D...A	<i>OpenProcess(..., 0, CsrGetProcessId³())</i> → no error if SeDebugPrivilege acquired
Timing	.S....	comparison of two <i>RDTSC</i> , inlined <i>GetTickCount⁴</i> , GS resets, ...
Timing API	.S...A	comparison of two APIs like <i>GetTickCount</i> , <i>GetSystemTime</i> , <i>QueryPerformanceCounter</i> , ...

Exceptions tricks (in the exception handler)

jmp	...X	change resume address via <i>Context.regEIP</i> (Context+B8)
step	...X	step next instruction and re-trigger via setting <i>TF</i> in <i>Context.EFlags</i> (Context+C0)
hwbp	...X	set or detects hardware breakpoint via <i>Context.dr*</i> (Context+04/+18)
higher	...X	overwrite higher handler and trigger exception ([esp+18])
return	...X	overwrite return address in stack → context re-loading is skipped ([esp+24])

Ollydbg (1.1) specific

esi	D.....	esi = -1 on startup under ollydbg, not in general
FPU	D.....	Display FFFFFFFF FFFFFFFF C0/40 3D as float → crash
OdbgStr	D...A	<i>OutputDebugStringA("%s%s")</i> → crash

VmWare specific

backdoor	D.....	in 'VMXh', 'VX' → exception if not present, else modified eax and ebx
sidt	D.....	[operand + 5] == e8 or ff if present
sldt	D.....	result != 0 if present
str	D.....	result == 4000h if present

Flags

- **anti Debugger** detects if debugger is attached, or if specific tool is used
- **anti Stepping** detects the program is not running (ex, not at full speed)
- **anti Emulator** detects the program is running under an emulator - via a non documented or complex behavior
- **get eIp** provides a way to retrieve current EIP
- **eXception based** relies on exceptions (Vectored, Structured, Unhandled filter)
- **Api-based** relies on an API call

Reminders

- TF is used by a debugger for stepping: set TF, an exception will be triggered after next execution is stepped
- TEB is at `fs:[18]`
- the PEB is accessible directly (`fs:[30]`) or via `TEB.EnvironmentPointer` (`[fs:[18] + 30]`)
- `LastError` is accessible via `TEB.LastErrorValue` (`[fs:[18] + 34]`) → `GetLastError` is inlinable

Notes

1. `70 = FLG_HEAP_ENABLE_TAIL_CHECK | FLG_HEAP_ENABLE_FREE_CHECK | FLG_HEAP_VALIDATE_PARAMETERS`
2. `FFFFFFFF` (constant value)
3. `dword[7C980380]` (inlineable)
4. `dword[7FFE0000] * dword[7FFE0004] >> 24` (inlineable)

Ange Albertini, 2010, cc by 3.0 <http://corkami.googlecode.com/files/cm.pdf> source