



Reverse Engineering with Hardware Debuggers

11 Mar 10

JASON RABER and JASON CHEATHAM

ATSPI Assessment Science Team

RYTA

Air Force Research Laboratory



Outline

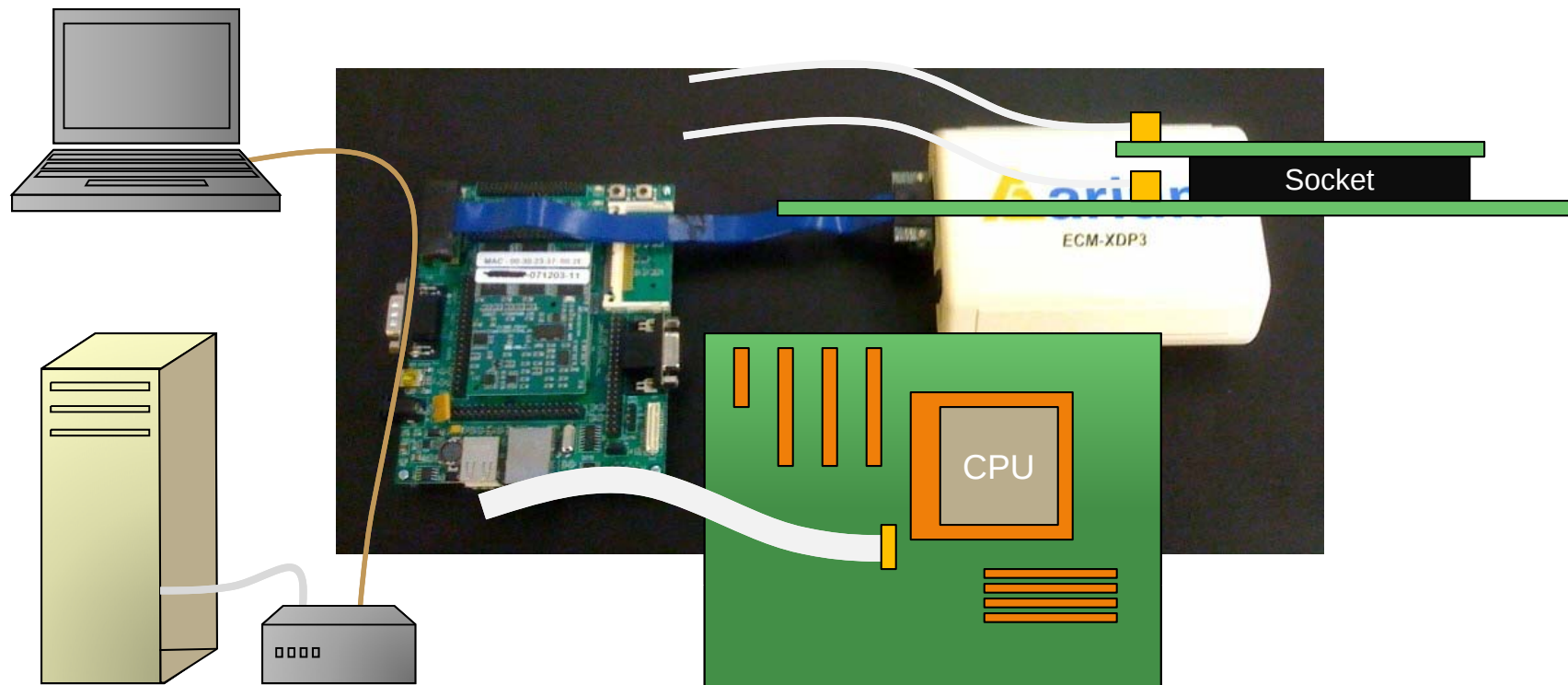


- **Architecture**
- **Breakpoints**
- **Macros**
- **Hypervisors**



The Hard Way

Hardware debugger / in-target probe (ITP) / in-circuit emulator (ICE)





Friends with Benefits



- **Hardware debuggers can see almost everything**
- **They live outside the OS, so even kernel mode rootkits can't hide**
- **Rewriting firmware**
- **They're OS independent**
 - Just needs a compatible x86 processor



The Softer Side

SourcePoint - x86 Family 6 Model F(M) - C:\Users\Jason\Documents\(\x86)\Arium\SourcePoint-IA\WinHyp.pri - [Memory (P1): 0008:F7589162]

File Edit View Processor Options Memory Window Help

Memory (P0): 0008:F7589162

0008:F7589162 73FB3B11
0008:F7589166 1405F622
0008:F758916A 01F758AB
0008:F758916E 5251BE74
0008:F7589172 696A5657
0008:F7589176 AB0C35FF
0008:F758917A 35FF7F58
0008:F758917E F758AB08
0008:F7589182 FFE473E9
0008:F7589186 8AA5E2FF
0008:F758918A 508DF5D
0008:F758918E 85C1031C
0008:F7589192 0F9974D2
0008:F7589196 5051CBB6
0008:F758919A 1075FF52
0008:F758919E FFFA95E9
0008:F75891A2 008AF4FF
0008:F75891A6 143D0000
0008:F75891AA 0F0004D0
0008:F75891B2 0C478BFF
0008:F75891B6 830478B8
0008:F75891BA 15732CF7
0008:F75891BE AB1405F6
0008:F75891C2 0F01F758
0008:F75891C6 FFFF6384
0008:F75891CA 6C6A56FF
0008:F75891CE FFFF4AF9

Memory (P1): 0008:F7589162

0008:F7589162 11 3B FB 73 22 F6 05 14 AB 58 F7 01 74 BE 51 52 .as"o. \X- thQR
0008:F7589166 57 56 6A 69 FF 35 0C AB 58 F7 FF 35 08 AB 58 F7 WViys. \X-p5. \X-
0008:F758916A E8 73 E4 FF FF EB A5 8A D0 FF 8D 50 1C 03 C1 85 essayeWjlyP. A
0008:F758916E D2 74 99 0F B6 CB 51 50 52 FF 75 10 E8 95 FA FF 0t1 EQPRyu. eIuy
0008:F7589172 FF E9 8A 00 00 0D 14 D0 04 00 0F 85 7B FF FF yel - D. .l(yv
0008:F7589176 FF 8B 47 0C 8B 7B 04 83 FF 2C 73 15 F6 05 14 AB yfC l(l(y. s. o. \X-
0008:F758917A 58 F7 01 0F 84 63 FF FF FF 56 6A 6C E9 4A FF FF X- .lcyvVjleJyy
0008:F758917E FF 8A 58 07 8B 48 18 0F B6 D3 8D 34 11 3B FE BE yIX lH. 704. .p
0008:F7589182 09 A7 58 F7 73 A4 F6 05 14 AB 58 F7 01 0F 84 39 .SX-SX. \X- .l9
0008:F7589186 FF FF 51 52 57 56 6A 6D E9 76 FF FF FF 3D 14 yyyQRWjleJyy
0008:F758918A D0 04 00 75 0E F6 05 14 AB 58 F7 01 74 22 56 6A B. u. o. \X- t'Vj
0008:F758918E 6F EB 0C F6 05 14 AB 58 F7 01 74 14 56 6A 70 FF 0e. o. \X- t. Vjpy
0008:F7589192 35 0C AB 58 F7 FF 35 08 AB 58 F7 E8 D4 E0 FF FF 5. \X-p5. \X-oayy
0008:F7589196 8B 75 0C 8B 45 10 6A 18 56 83 C0 40 50 FF 15 EC lu. lE. j. VjleJyy
0008:F758919A A1 58 F7 8B 48 18 5F 5E 5B C9 C2 0C 00 CC CC lE-lE. [EA. lll
0008:F758919E CC CC 8B FF 55 8B EC 53 56 8B 75 08 33 DB 39 5E llyVjleJyy. 309?
0008:F75891A2 58 57 BF 11 00 00 80 0F 85 CF 00 00 53 53 53 XVL. l. l. l. SSS
0008:F75891A6 8D 46 60 53 50 FF 15 AC A1 58 F7 8B F8 3B FB 0F F SFy. -lX-lE. u.
0008:F75891AA 8C A7 00 00 39 5E 5B 0F 85 9E 00 00 56 E8 lS. 9X l. l. Va
0008:F75891AE 94 57 00 00 8B F8 3B FB 7D 28 F6 05 14 AB 58 F7 lV. lE. u. j. o. \X-
0008:F75891B2 01 0F 84 85 00 00 00 57 68 08 A7 58 F7 6A 19 FF l. l. Wh. SX-j. y
0008:F75891B6 35 0C AB 58 F7 FF 35 08 AB 58 F7 E8 0A E3 FF FF 5. \X-p5. \X-e. 3yy
0008:F75891BA EB 6A 56 E8 C0 56 00 00 8B F8 3B FB 7D 13 F6 05 eJVeA. lE. u. j. o.
0008:F75891BE 14 AB 58 F7 01 74 55 57 68 08 A7 58 F7 6A 1A EE \X- .lVWh. SX-j. e
0008:F75891C2 CE 56 E8 61 54 00 00 8B F8 3B FB 7D 13 F6 05 14 lVeaT. lE. u. j. o.
0008:F75891C6 AB 58 F7 01 74 36 57 68 08 A7 58 F7 6A 1B EB AF \X- t6Wh. SX-j. e

General Registers (P1)

Legacy	Name	Value
General	EAX	7617CC30
Floating Point	EBX	F7717C70
Segment	ECX	F7717C70
Control	EDX	00000027
Debug	EBP	F78AED50
MMX	ESI	F7717C50
XMM	EDI	89BD5E88
XMM- DP	ESP	F78AED34
XMM- Int	CS	0008
MSR	DS	0023
IA-32e	SS	0010
User	ES	0023
	FS	0030
	GS	0000
	EIP	F76F9162
	EFLAGS	00010206

Breakpoints

Identifier	Address	Attributes
NtCreateProcess	8058B7CD	Execute (All Process)
NtOpenFile	8056C4A1	Execute (All Process)
SystemCall	7C90E40F	Execute (All Process)
event004	00350EEB0P	Execute (Software)

0008:F76F915D 895104 mov dword ptr [ecx]+04
0008:F76F9160 FB stl
0008:F76F9161 F4 hlt
0008:F76F9162 6A00 push 00000000
0008:F76F9163 59 pop ecx
0008:F76F9164 E841040000 call near32 ptr f76f95e
0008:F76F9165 59 pop ecx
0008:F76F9166 894108 mov dword ptr [ecx]+08
0008:F76F9167 89510C mov dword ptr [ecx]+0C
0008:F76F9168 33C0 xor eax, eax
0008:F76F9169 C3 retn
0008:F76F916A 90 nop
0008:F76F916B 51 push ecx
0008:F76F916C 6A00 push 00000000
0008:F76F916D 59 pop ecx
0008:F76F916E 59 pop ecx
0008:F76F916F 59 pop ecx
0008:F76F9170 33C0 xor eax, eax
0008:F76F9171 C3 retn
0008:F76F9172 90 nop
0008:F76F9173 51 push ecx
0008:F76F9174 6A00 push 00000000
0008:F76F9175 59 pop ecx
0008:F76F9176 59 pop ecx
0008:F76F9177 59 pop ecx
0008:F76F9178 33C0 xor eax, eax
0008:F76F9179 C3 retn
0008:F76F917A 90 nop
0008:F76F917B 51 push ecx
0008:F76F917C 6A00 push 00000000
0008:F76F917D 59 pop ecx
0008:F76F917E 59 pop ecx
0008:F76F917F 59 pop ecx
0008:F76F9180 33C0 xor eax, eax
0008:F76F9181 C3 retn
0008:F76F9182 90 nop
0008:F76F9183 51 push ecx
0008:F76F9184 6A00 push 00000000
0008:F76F9185 59 pop ecx
0008:F76F9186 59 pop ecx
0008:F76F9187 59 pop ecx
0008:F76F9188 33C0 xor eax, eax
0008:F76F9189 C3 retn
0008:F76F918A 90 nop
0008:F76F918B 51 push ecx
0008:F76F918C 6A00 push 00000000
0008:F76F918D 59 pop ecx
0008:F76F918E 59 pop ecx
0008:F76F918F 59 pop ecx
0008:F76F9190 33C0 xor eax, eax
0008:F76F9191 C3 retn
0008:F76F9192 90 nop
0008:F76F9193 51 push ecx
0008:F76F9194 6A00 push 00000000
0008:F76F9195 59 pop ecx
0008:F76F9196 59 pop ecx
0008:F76F9197 59 pop ecx
0008:F76F9198 33C0 xor eax, eax
0008:F76F9199 C3 retn
0008:F76F919A 90 nop
0008:F76F919B 51 push ecx
0008:F76F919C 6A00 push 00000000
0008:F76F919D 59 pop ecx
0008:F76F919E 59 pop ecx
0008:F76F919F 59 pop ecx
0008:F76F91A0 33C0 xor eax, eax
0008:F76F91A1 C3 retn
0008:F76F91A2 90 nop
0008:F76F91A3 51 push ecx
0008:F76F91A4 6A00 push 00000000
0008:F76F91A5 59 pop ecx
0008:F76F91A6 59 pop ecx
0008:F76F91A7 59 pop ecx
0008:F76F91A8 33C0 xor eax, eax
0008:F76F91A9 C3 retn
0008:F76F91AA 90 nop
0008:F76F91AB 51 push ecx
0008:F76F91AC 6A00 push 00000000
0008:F76F91AD 59 pop ecx
0008:F76F91AE 59 pop ecx
0008:F76F91AF 59 pop ecx
0008:F76F91B0 33C0 xor eax, eax
0008:F76F91B1 C3 retn
0008:F76F91B2 90 nop
0008:F76F91B3 51 push ecx
0008:F76F91B4 6A00 push 00000000
0008:F76F91B5 59 pop ecx
0008:F76F91B6 59 pop ecx
0008:F76F91B7 59 pop ecx
0008:F76F91B8 33C0 xor eax, eax
0008:F76F91B9 C3 retn
0008:F76F91BA 90 nop
0008:F76F91BB 51 push ecx
0008:F76F91BC 6A00 push 00000000
0008:F76F91BD 59 pop ecx
0008:F76F91BE 59 pop ecx
0008:F76F91BF 59 pop ecx
0008:F76F91C0 33C0 xor eax, eax
0008:F76F91C1 C3 retn
0008:F76F91C2 90 nop
0008:F76F91C3 51 push ecx
0008:F76F91C4 6A00 push 00000000
0008:F76F91C5 59 pop ecx
0008:F76F91C6 59 pop ecx
0008:F76F91C7 59 pop ecx
0008:F76F91C8 33C0 xor eax, eax
0008:F76F91C9 C3 retn
0008:F76F91CA 90 nop
0008:F76F91CB 51 push ecx
0008:F76F91CC 6A00 push 00000000
0008:F76F91CD 59 pop ecx
0008:F76F91CE 59 pop ecx
0008:F76F91CF 59 pop ecx
0008:F76F91D0 33C0 xor eax, eax
0008:F76F91D1 C3 retn
0008:F76F91D2 90 nop
0008:F76F91D3 51 push ecx
0008:F76F91D4 6A00 push 00000000
0008:F76F91D5 59 pop ecx
0008:F76F91D6 59 pop ecx
0008:F76F91D7 59 pop ecx
0008:F76F91D8 33C0 xor eax, eax
0008:F76F91D9 C3 retn
0008:F76F91DA 90 nop
0008:F76F91DB 51 push ecx
0008:F76F91DC 6A00 push 00000000
0008:F76F91DD 59 pop ecx
0008:F76F91DE 59 pop ecx
0008:F76F91DF 59 pop ecx
0008:F76F91E0 33C0 xor eax, eax
0008:F76F91E1 C3 retn
0008:F76F91E2 90 nop
0008:F76F91E3 51 push ecx
0008:F76F91E4 6A00 push 00000000
0008:F76F91E5 59 pop ecx
0008:F76F91E6 59 pop ecx
0008:F76F91E7 59 pop ecx
0008:F76F91E8 33C0 xor eax, eax
0008:F76F91E9 C3 retn
0008:F76F91EA 90 nop
0008:F76F91EB 51 push ecx
0008:F76F91EC 6A00 push 00000000
0008:F76F91ED 59 pop ecx
0008:F76F91EE 59 pop ecx
0008:F76F91EF 59 pop ecx
0008:F76F91F0 33C0 xor eax, eax
0008:F76F91F1 C3 retn
0008:F76F91F2 90 nop
0008:F76F91F3 51 push ecx
0008:F76F91F4 6A00 push 00000000
0008:F76F91F5 59 pop ecx
0008:F76F91F6 59 pop ecx
0008:F76F91F7 59 pop ecx
0008:F76F91F8 33C0 xor eax, eax
0008:F76F91F9 C3 retn
0008:F76F91FA 90 nop
0008:F76F91FB 51 push ecx
0008:F76F91FC 6A00 push 00000000
0008:F76F91FD 59 pop ecx
0008:F76F91FE 59 pop ecx
0008:F76F91FF 59 pop ecx
0008:F76F9200 33C0 xor eax, eax
0008:F76F9201 C3 retn
0008:F76F9202 90 nop
0008:F76F9203 51 push ecx
0008:F76F9204 6A00 push 00000000
0008:F76F9205 59 pop ecx
0008:F76F9206 59 pop ecx
0008:F76F9207 59 pop ecx
0008:F76F9208 33C0 xor eax, eax
0008:F76F9209 C3 retn
0008:F76F920A 90 nop
0008:F76F920B 51 push ecx
0008:F76F920C 6A00 push 00000000
0008:F76F920D 59 pop ecx
0008:F76F920E 59 pop ecx
0008:F76F920F 59 pop ecx
0008:F76F9210 33C0 xor eax, eax
0008:F76F9211 C3 retn
0008:F76F9212 90 nop
0008:F76F9213 51 push ecx
0008:F76F9214 6A00 push 00000000
0008:F76F9215 59 pop ecx
0008:F76F9216 59 pop ecx
0008:F76F9217 59 pop ecx
0008:F76F9218 33C0 xor eax, eax
0008:F76F9219 C3 retn
0008:F76F921A 90 nop
0008:F76F921B 51 push ecx
0008:F76F921C 6A00 push 00000000
0008:F76F921D 59 pop ecx
0008:F76F921E 59 pop ecx
0008:F76F921F 59 pop ecx
0008:F76F9220 33C0 xor eax, eax
0008:F76F9221 C3 retn
0008:F76F9222 90 nop
0008:F76F9223 51 push ecx
0008:F76F9224 6A00 push 00000000
0008:F76F9225 59 pop ecx
0008:F76F9226 59 pop ecx
0008:F76F9227 59 pop ecx
0008:F76F9228 33C0 xor eax, eax
0008:F76F9229 C3 retn
0008:F76F922A 90 nop
0008:F76F922B 51 push ecx
0008:F76F922C 6A00 push 00000000
0008:F76F922D 59 pop ecx
0008:F76F922E 59 pop ecx
0008:F76F922F 59 pop ecx
0008:F76F9230 33C0 xor eax, eax
0008:F76F9231 C3 retn
0008:F76F9232 90 nop
0008:F76F9233 51 push ecx
0008:F76F9234 6A00 push 00000000
0008:F76F9235 59 pop ecx
0008:F76F9236 59 pop ecx
0008:F76F9237 59 pop ecx
0008:F76F9238 33C0 xor eax, eax
0008:F76F9239 C3 retn
0008:F76F923A 90 nop
0008:F76F923B 51 push ecx
0008:F76F923C 6A00 push 00000000
0008:F76F923D 59 pop ecx
0008:F76F923E 59 pop ecx
0008:F76F923F 59 pop ecx
0008:F76F9240 33C0 xor eax, eax
0008:F76F9241 C3 retn
0008:F76F9242 90 nop
0008:F76F9243 51 push ecx
0008:F76F9244 6A00 push 00000000
0008:F76F9245 59 pop ecx
0008:F76F9246 59 pop ecx
0008:F76F9247 59 pop ecx
0008:F76F9248 33C0 xor eax, eax
0008:F76F9249 C3 retn
0008:F76F924A 90 nop
0008:F76F924B 51 push ecx
0008:F76F924C 6A00 push 00000000
0008:F76F924D 59 pop ecx
0008:F76F924E 59 pop ecx
0008:F76F924F 59 pop ecx
0008:F76F9250 33C0 xor eax, eax
0008:F76F9251 C3 retn
0008:F76F9252 90 nop
0008:F76F9253 51 push ecx
0008:F76F9254 6A00 push 00000000
0008:F76F9255 59 pop ecx
0008:F76F9256 59 pop ecx
0008:F76F9257 59 pop ecx
0008:F76F9258 33C0 xor eax, eax
0008:F76F9259 C3 retn
0008:F76F925A 90 nop
0008:F76F925B 51 push ecx
0008:F76F925C 6A00 push 00000000
0008:F76F925D 59 pop ecx
0008:F76F925E 59 pop ecx
0008:F76F925F 59 pop ecx
0008:F76F9260 33C0 xor eax, eax
0008:F76F9261 C3 retn
0008:F76F9262 90 nop
0008:F76F9263 51 push ecx
0008:F76F9264 6A00 push 00000000
0008:F76F9265 59 pop ecx
0008:F76F9266 59 pop ecx
0008:F76F9267 59 pop ecx
0008:F76F9268 33C0 xor eax, eax
0008:F76F9269 C3 retn
0008:F76F926A 90 nop
0008:F76F926B 51 push ecx
0008:F76F926C 6A00 push 00000000
0008:F76F926D 59 pop ecx
0008:F76F926E 59 pop ecx
0008:F76F926F 59 pop ecx
0008:F76F9270 33C0 xor eax, eax
0008:F76F9271 C3 retn
0008:F76F9272 90 nop
0008:F76F9273 51 push ecx
0008:F76F9274 6A00 push 00000000
0008:F76F9275 59 pop ecx
0008:F76F9276 59 pop ecx
0008:F76F9277 59 pop ecx
0008:F76F9278 33C0 xor eax, eax
0008:F76F9279 C3 retn
0008:F76F927A 90 nop
0008:F76F927B 51 push ecx
0008:F76F927C 6A00 push 00000000
0008:F76F927D 59 pop ecx
0008:F76F927E 59 pop ecx
0008:F76F927F 59 pop ecx
0008:F76F9280 33C0 xor eax, eax
0008:F76F9281 C3 retn
0008:F76F9282 90 nop
0008:F76F9283 51 push ecx
0008:F76F9284 6A00 push 00000000
0008:F76F9285 59 pop ecx
0008:F76F9286 59 pop ecx
0008:F76F9287 59 pop ecx
0008:F76F9288 33C0 xor eax, eax
0008:F76F9289 C3 retn
0008:F76F928A 90 nop
0008:F76F928B 51 push ecx
0008:F76F928C 6A00 push 00000000
0008:F76F928D 59 pop ecx
0008:F76F928E 59 pop ecx
0008:F76F928F 59 pop ecx
0008:F76F9290 33C0 xor eax, eax
0008:F76F9291 C3 retn
0008:F76F9292 90 nop
0008:F76F9293 51 push ecx
0008:F76F9294 6A00 push 00000000
0008:F76F9295 59 pop ecx
0008:F76F9296 59 pop ecx
0008:F76F9297 59 pop ecx
0008:F76F9298 33C0 xor eax, eax
0008:F76F9299 C3 retn
0008:F76F929A 90 nop
0008:F76F929B 51 push ecx
0008:F76F929C 6A00 push 00000000
0008:F76F929D 59 pop ecx
0008:F76F929E 59 pop ecx
0008:F76F929F 59 pop ecx
0008:F76F92A0 33C0 xor eax, eax
0008:F76F92A1 C3 retn
0008:F76F92A2 90 nop
0008:F76F92A3 51 push ecx
0008:F76F92A4 6A00 push 00000000
0008:F76F92A5 59 pop ecx
0008:F76F92A6 59 pop ecx
0008:F76F92A7 59 pop ecx
0008:F76F92A8 33C0 xor eax, eax
0008:F76F92A9 C3 retn
0008:F76F92AA 90 nop
0008:F76F92AB 51 push ecx
0008:F76F92AC 6A00 push 00000000
0008:F76F92AD 59 pop ecx
0008:F76F92AE 59 pop ecx
0008:F76F92AF 59 pop ecx
0008:F76F92B0 33C0 xor eax, eax
0008:F76F92B1 C3 retn
0008:F76F92B2 90 nop
0008:F76F92B3 51 push ecx
0008:F76F92B4 6A00 push 00000000
0008:F76F92B5 59 pop ecx
0008:F76F92B6 59 pop ecx
0008:F76F92B7 59 pop ecx
0008:F76F92B8 33C0 xor eax, eax
0008:F76F92B9 C3 retn
0008:F76F92BA 90 nop
0008:F76F92BB 51 push ecx
0008:F76F92BC 6A00 push 00000000
0008:F76F92BD 59 pop ecx
0008:F76F92BE 59 pop ecx
0008:F76F92BF 59 pop ecx
0008:F76F92C0 33C0 xor eax, eax
0008:F76F92C1 C3 retn
0008:F76F92C2 90 nop
0008:F76F92C3 51 push ecx
0008:F76F92C4 6A00 push 00000000
0008:F76F92C5 59 pop ecx
0008:F76F92C6 59 pop ecx
0008:F76F92C7 59 pop ecx
0008:F76F92C8 33C0 xor eax, eax
0008:F76F92C9 C3 retn
0008:F76F92CA 90 nop
0008:F76F92CB 51 push ecx
0008:F76F92CC 6A00 push 00000000
0008:F76F92CD 59 pop ecx
0008:F76F92CE 59 pop ecx
0008:F76F92CF 59 pop ecx
0008:F76F92D0 33C0 xor eax, eax
0008:F76F92D1 C3 retn
0008:F76F92D2 90 nop
0008:F76F92D3 51 push ecx
0008:F76F92D4 6A00 push 00000000
0008:F76F92D5 59 pop ecx
0008:F76F92D6 59 pop ecx
0008:F76F92D7 59 pop ecx
0008:F76F92D8 33C0 xor eax, eax
0008:F76F92D9 C3 retn
0008:F76F92DA 90 nop
0008:F76F92DB 51 push ecx
0008:F76F92DC 6A00 push 00000000
0008:F76F92DD 59 pop ecx
0008:F76F92DE 59 pop ecx
0008:F76F92DF 59 pop ecx
0008:F76F92E0 33C0 xor eax, eax
0008:F76F92E1 C3 retn
0008:F76F92E2 90 nop
0008:F76F92E3 51 push ecx
0008:F76F92E4 6A00 push 00000000
0008:F76F92E5 59 pop ecx
0008:F76F92E6 59 pop ecx
0008:F76F92E7 59 pop ecx
0008:F76F92E8 33C0 xor eax, eax
0008:F76F92E9 C3 retn
0008:F76F92EA 90 nop
0008:F76F92EB 51 push ecx
0008:F76F92EC 6A00 push 00000000
0008:F76F92ED 59 pop ecx
0008:F76F92EE 59 pop ecx
0008:F76F92EF 59 pop ecx
0008:F76F92F0 33C0 xor eax, eax
0008:F76F92F1 C3 retn
0008:F76F92F2 90 nop
0008:F76F92F3 51 push ecx
0008:F76F92F4 6A00 push 00000000
0008:F76F92F5 59 pop ecx
0008:F76F92F6 59 pop ecx
0008:F76F92F7 59 pop ecx
0008:F76F92F8 33C0 xor eax, eax
0008:F76F92F9 C3 retn
0008:F76F92FA 90 nop
0008:F76F92FB 51 push ecx
0008:F76F92FC 6A00 push 00000000
0008:F76F92FD 59 pop ecx
0008:F76F92FE 59 pop ecx
0008:F76F92FF 59 pop ecx
0008:F76F9300 33C0 xor eax, eax
0008:F76F9301 C3 retn
0008:F76F9302 90 nop
0008:F76F9303 51 push ecx
0008:F76F9304 6A00 push 00000000
0008:F76F9305 59 pop ecx
0008:F76F9306 59 pop ecx
0008:F76F9307 59 pop ecx
0008:F76F9308 33C0 xor eax, eax
0008:F76F9309 C3 retn
0008:F76F930A 90 nop
0008:F76F930B 51 push ecx
0008:F76F930C 6A00 push 00000000
0008:F76F930D 59 pop ecx
0008:F76F930E 59 pop ecx
0008:F76F930F 59 pop ecx
0008:F76F9310 33C0 xor eax, eax
0008:F76F9311 C3 retn
0008:F76F9312 90 nop
0008:F76F9313 51 push ecx
0008:F76F9314 6A00 push 00000000
0008:F76F9315 59 pop ecx
0008:F76F9316 59 pop ecx
0008:F76F9317 59 pop ecx
0008:F76F9318 33C0 xor eax, eax
0008:F76F9319 C3 retn
0008:F76F931A 90 nop
0008:F76F931B 51 push ecx
0008:F76F931C 6A00 push 00000000
0008:F76F931D 59 pop ecx
0008:F76F931E 59 pop ecx
0008:F76F931F 59 pop ecx
0008:F76F9320



Outline



- Architecture
- Breakpoints
- Macros
- Hypervisors



Ways to Break Things



- **Hardware Breakpoints**
 - DR registers
- **Software Breakpoints**
 - ICEBP (0xF1) instruction + DR7 bit 12
- **Infinite loops**
 - Steal a couple bytes and replace with 0xEBFE



Infinite Breakpoints

- **EB FE is a jump to the same address (jmp \$)**
 - Inject the infinite loop (0xEBFE) into the application or driver
 - Halt the CPU when the system freezes, and there you are
- **They're very easy to use with an ICE**
 - No worries about freezing the system
 - Don't have to deal with virtual memory
- **Checksums can detect these, so place them carefully**



Outline



- Architecture
- Breakpoints
- **Macros**
- Hypervisors



More Power...



- **Macros can make an emulator very powerful**
 - Implement complex or repetitive tasks
 - Detailed control of ICE
- **SourcePoint uses a C like scripting language**
 - Variables, functions, control flow
 - Types have well-defined widths
 - ord1, int4, real8, ...
 - Control statements for ICE

```
define proc pcrange(startaddr,
    endaddr)
    define ord4 startaddr
    define ord4 endaddr
{
    while (1) {
        if (EIP >= startaddr &&
            EIP <= endaddr) {
            break
        } else {
            step 4
        }
    }
}
```



Range Breakpoint

SourcePoint - x86 Family 6 Model F (M) - C:\Users\jason\Documents\SourcePoint\WinVlp.p...

File Edit View Processor Options Command Window Help

Breakpoints

Identifier	Address	Attributes
------------	---------	------------

Edit... Add... Remove Remove All Enable Disable All

Viewpoint - (Go/Stop = All Processors)

Name	Description	Status
P0	x86 Family 6 Model F (M)	Running
P1	x86 Family 6 Model F (M)	Running

Code (P0): (32-bit) Tracking IP: 001B:00000000 - 001B:FFFFFFF

Address	Disassembly	Comment
001B:0040106A	6A12	push 00000012
001B:0040106C	6800104000	push 00401000
001B:00401071	FF1500A04000	call dword ptr [0040a000]
001B:00401077	68AF000000	push 000000af
001B:0040107C	6A12	push 00000012
001B:0040107E	6800104000	push 00401000
001B:00401083	E898FFFFFF	call near32 ptr 00401020
001B:00401088	83C40C	add esp,0000000c
001B:0040108B	E870FFFFFF	call near32 ptr 00401000
001B:00401090	33C0	xor eax,eax
001B:00401092	8BE5	mov esp,ebp
001B:00401094	5D	pop ebp
001B:00401095	C3	ret

001B:00401060 Disassembly Go Cursor Set Break Track IP View IP

Code (P1): (32-bit) Tracking IP: 0008:00000000 - 0008:FFFFFFF

Address	Disassembly	Comment
0008:F758916A	894108	mov dword ptr [ecx],eax
0008:F758916D	89510C	mov dword ptr [ecx],eax
0008:F7589170	33C0	xor eax,eax
0008:F7589172	C3	ret
0008:F7589173	90	nop
0008:F7589174	51	push ecx
0008:F7589175	6A00	push 00000000
0008:F7589177	E82E040000	call near32 ptr f7589170
0008:F758917C	8B0C24	mov ecx,dword ptr [ecx]
0008:F758917F	8901	mov dword ptr [ecx],eax
0008:F7589181	895104	mov dword ptr [ecx],eax
0008:F7589184	F6411001	test byte ptr [ecx],01
0008:F7589188	7527	jne short ptr f758918f

0008:F7589162 Disassembly Go Cursor Set Break Track IP

Command

P1>
P1>

Log

Date	Time	Component	Message
00000080	00000000	00000000	00000000
00000090	00000000	00000000	00000000
00000096	002F0000	00000000	00000000
00000010	00000005	00000000	00000000
00000020	001A0000	001E0000	00000000
00000030	00000000	00000000	00000000
00000040	00000000	00000000	00000000
00000050	00000000	00000000	00000000
00000060	00000000	00000000	00000000
00000070	00000000	00000000	00000000
00000080	00000000	00000000	00000000
00000090	00000000	00000000	00000000

F1:Help, F5:Go, Shift+F5:Stop, F8:Step Into, F10:Step Over, Shift+F12:Reset

P1 1F: Running Halt Mode



Run Trace



```
flist ("tracelog.txt", 1)
define ord4 lasteip = EIP
softremove
while (1) {
    if (EIP >= 0xC0000000 && lasteip < 0xC0000000) {
        softbreak = location=lasteip+2
        softbreak = location=lasteip+3
        softbreak = location=lasteip+4
        softbreak = location=lasteip+5
        softbreak = location=lasteip+6
        lasteip = EIP
        go
    }

    if (EIP != endaddr) {
        asm eip
        printf("eax: "); eval eax virtual
        printf("ebx: "); eval ebx virtual
        printf("ecx: "); eval ecx virtual
        printf("edx: "); eval edx virtual
        printf("esi: "); eval esi virtual
        printf("edi: "); eval edi virtual
        printf("esp: "); eval esp virtual
        lasteip = EIP
        step
    } else {
        nolog
        stop
    }
}
```

The screenshot shows a debugger window with a blue title bar. The main area displays assembly instructions and register values. The first instruction is 'XOR ECX, ECX' at address 0073:08048243. The second instruction is 'TEST EAX, EAX' at address 0073:08048245. The register values are: esp: 007B:BFE0F2B0, eax: 007B:00000000, ebx: 007B:00000001, ecx: 007B:BFE0F4D4, edx: 007B:BFE0F4DC, esi: 007B:00000001, edi: 007B:00000000. At the bottom, a red message states 'Target is already stopped'.

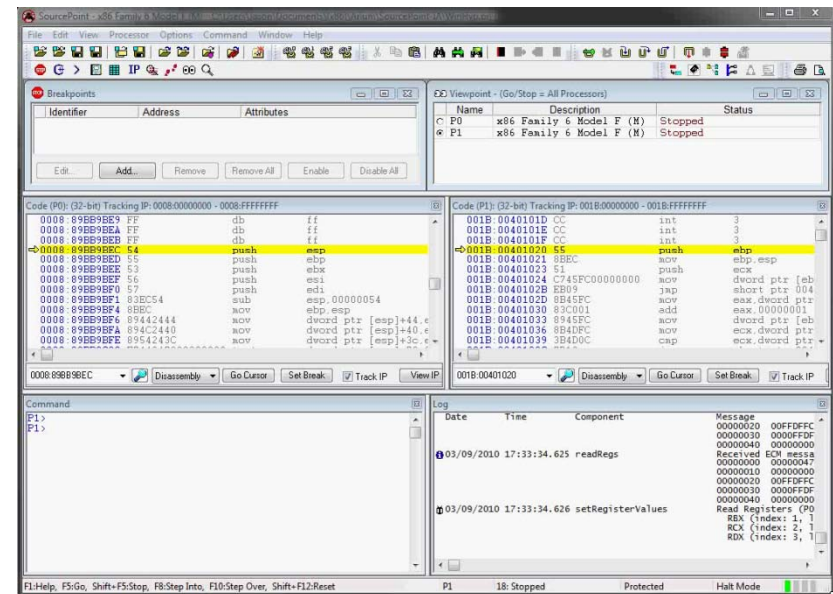
Address	Instruction	Operands
0073:08048243	XOR	ECX, ECX
0073:08048245	TEST	EAX, EAX



From the Outside In



- Emulator access is pretty raw, so we wrote some simple forensic macros
 - list_procs()
 - get_ssdt()
 - get_syscall_addr(name)
 - hook_syscall(name)
 - list_mods()





get_syscall_addr

```
define proc pointer get_syscall_addr(syscall_name)
  define nstring syscall_name
  {
    if (syscall_name == "NtCreateProcess")
      return ssdt_index_to_addr(0x30, 0)
    } else if (syscall_name == "NtCreateSe")
      return ssdt_index_to_addr(0x32, 0)
    } else {
      printf("Unknown syscall: %s", syscall_name)
      return 0
    }
  }
```

```
define proc pointer ssdt_index_to_addr(index, offset)
  {
    define ord1 index
    define ord2 offset
    return ssdt_index_to_addr(ord1, ord2)
  }
```

```
define proc pointer ssdt_index_to_addr(ord1, ord2)
  {
    define pointer ssdt_index_to_addr(ord1, ord2)
    define pointer ssdt_index_to_addr(ord1, ord2)
    define pointer ssdt_index_to_addr(ord1, ord2)
    return ssdt_index_to_addr(ord1, ord2)
  }
```

```
define proc pointer find_gdi_proc() {
  define pointer PsActiveProcessHead = get_head_proc()

  define pointer first_proc = flink_to_proc_addr(PsActiveProcessHead)

  define pointer current_flink = next_proc_flink(first_proc)
  define pointer current_proc = flink_to_proc_addr(current_flink)

  while (current_flink != PsActiveProcessHead) {
    define ord1 type = proc_type(current_proc)
    if (type != 0x03) { printf("Non-process type: %x\n", type); break; }

    define ord4 w32proc = proc_win32process(current_proc)

    if (w32proc != 0x0) return current_proc

    current_flink = next_proc_flink(current_proc)
    current_proc = flink_to_proc_addr(current_flink)
  }
}
```



Outline



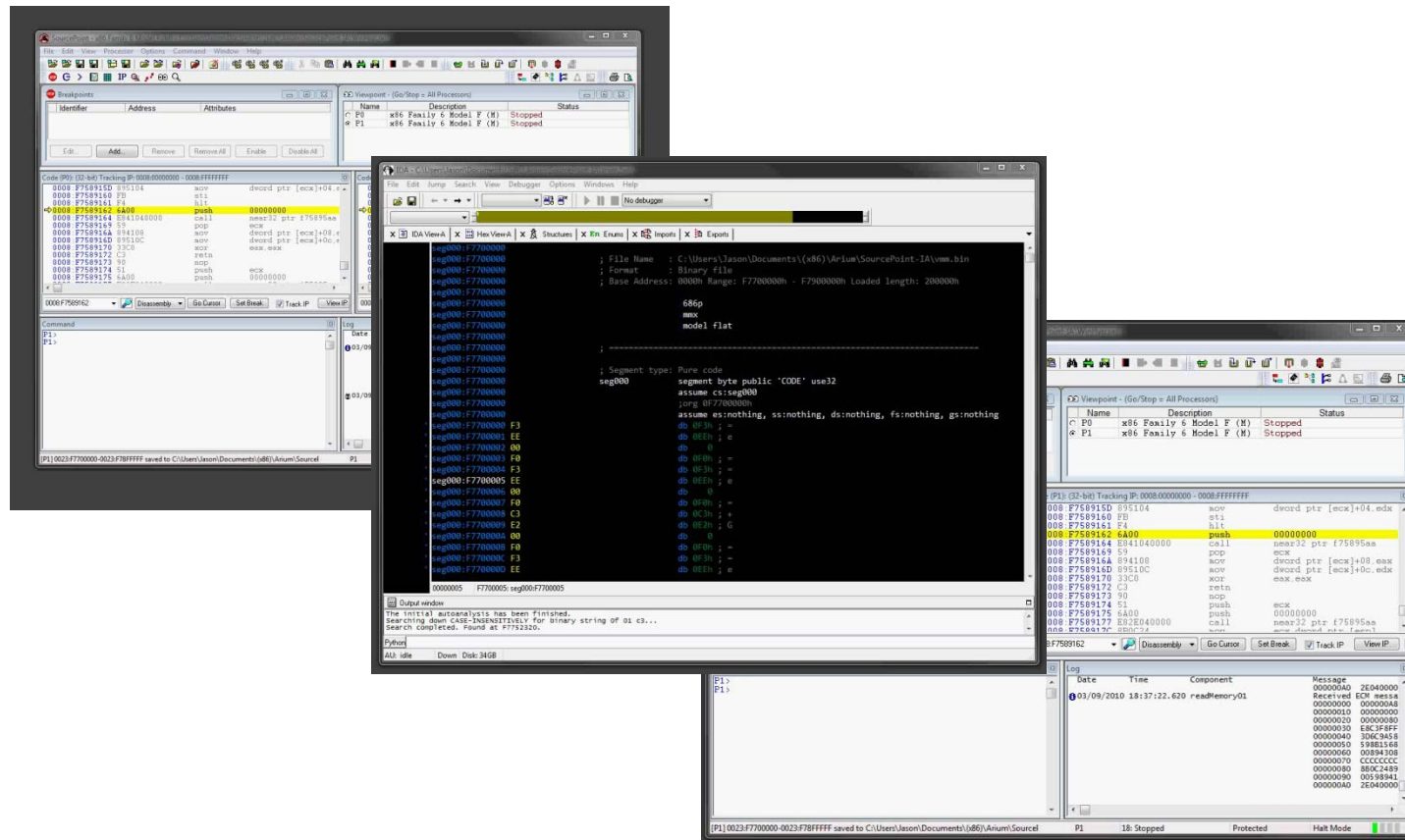
- **Architecture**
- **Breakpoints**
- **Macros**
- **Hypervisors**



Blue...Something



- **ICE runs below hypervisors**
- **You can go far with a few simple macros**
 - vmx_is_enabled
 - vmx_goto_resume
 - vmx_guest_eip
 - vmx_exit_reason





Nobody's Perfect



- You can't single step or trace across the ring -1/0 boundary
- Finding the hypervisor is tricky
- Newer VM instructions can cause problems



Summary

