Native Client: A sandbox for portable, untrusted x86 native code

Bennet Yee, David Sehr, Gregory Dardyk, J. Bradley Chen, Robert Muth, Tavis Ormandy, Shiki Okasaka, Neha Narula, Nicholas Fulgar (Google Inc.)

Dresden, 2010-04-27
Not yet Web 2.0
Native code execution

- Faster than interpreted code
- Make use of platform-specific assembly (e.g., SSE)
- Arbitrary code → Security threat

**NaCl**: framework to support **safe** execution of **x86 machine code** in a **sandbox**
Recap: Software Fault Isolation

• Robert Wahbe, 1993

• Plugins in sub-address spaces (segments)
  – Segment matching: check that plugin stays within sandbox
    • Mostly static checks
    • Additionally insert runtime checks
  – Address sandboxing
    • For each memory access fix upper bits of address to segment idx
  – System calls & system resource accesses $\rightarrow$ cross-domain RPC

• Limitations
  – RISC (extended to CISC: XFI, Erlingson 2006)
  – x86 register scarcity
Figure 2: The hypothetical photo application of Figure 1 with a trusted storage service.
Sandboxing native code

- **Outer sandbox:**
  - System-call monitoring

- **Inner sandbox**
  - Static checking at load-time
  - Dynamic runtime checks

- **Service runtime**
  - System-level interface
Static checking

- Reliable disassembly
  - All valid code within text segment
  - No self-modifying code

- No unsafe instructions
  - SYSENTER, INT, segment-related instructions, RET
  - Ring 0 instructions

- Control-flow integrity
  - Ensure each jmp goes to a valid instruction
Runtime checks

- Indirect jumps: `nacl_jump` and `%eax, 0xFFFFFFFFFe0 jmp *%eax`

- Use x86 segmentation to enforce sandbox
  - Restriction: x86/32bit

- Disallow (asynchronous) hardware exceptions
  - Would need to copy with stack segment, which is invalidated during NaCl execution
Service runtime

- **Unrestricted code**
- **System call trampolines**
  - save/restore segments
  - 32-byte aligned
  - one per system call
- **Springboard**
  - Allow calls into NaCl modules
  - Potentially unrestricted
  - Start with HLT
- **IMC sockets**
  - Datagram-based
  - Higher-level protocols on top
Evaluation

- Modified GCC 4.2.2 + Binutils 2.18

- SPEC2000: average 5%., top 12% overhead in NaCl mode

- Near-native performance for
  - Computer graphics
  - H.264 decoding
  - Quake (yeah!)

- Going into Google Chrome
“This is my tentative endorsement, that, yes, Native Client could actually win

... 

but only if they lock Tavis Ormandy in a room for a year or two

... 

and I'm worried about the outer sandbox, so you should be too.”
Discussion

• Hack it?
  – Return-oriented programming works for fixed-length RISC instruction sets.
  – Doing harm depends on configuration of outer sandbox.