Microsoft's NGSCB

Trustworthy Computing?

Introduction

- Next Generation Secure Computing Base (NGSCB), formally called Palladium, is a Microsoft initiative to significantly change PC hardware and software to support new security features
- This initiative is directly related to the work of an industry group call the Trusted Computer Group (TCG)

Topics

- Provide some background about the NGSCB initiative
- The four design goals for NGSCB
- The NSGCB hardware changes
- Nexus—the new software kernel
- Provide some historical context and rationale for the NGSCB architecture
- Begin to consider attack vectors

Trustworthy vs. Trusted

- Trusted System--one which can break the security policy
- Trustworthy—”make the computing ecosystem sufficiently *trustworthy* that people don't worry about its fallibility or unreliability the way they do today...for people to be as comfortable using devices powered by computers and software as they are today using a device that is powered by electricity”

Societal/Economic Implications

- Negatives of successful (secure) NGSCB not considered here
- Visit Ross Anderson's site for consideration of negative impact of Digital Rights Management and overall economic badness
- "Treacherous computing" ... Richard Stallman, Free Software Foundation

Raise the Security Bar/ New Markets

- Microsoft says they want to stop nearly all software, and some limited hardware assisted, threats to a Windows based computer
- “revitalize the PC ecosystem by enabling a new generation of hardware and software products”
- New markets? NGSCB will require new hardware (new CPU chips, new motherboards, and new specialized chips to provide the crypto/security functions)

NGSCB Design Goals

- **Four Fundamentals:**
  - Strong Process Isolation
  - Sealed Storage
  - Attestation
  - Secure Paths to the User

Secure Process Isolation

- *The protected operating environment isolates a secure area of memory that is used to process data with higher security requirements.*
  - Windows currently provides separation by using ring 0 and ring 3 processor mode isolation
  - Windows, and other operating systems, also make use of virtual memory segmentation to separate processes from each other and from the operating system

Secure Process Isolation-cont'd

- Microsoft makes a remarkable admission that "Using the current memory scheme, only virtual memory protection is achievable, and it is relatively easy for an attacker to add malicious programs to both the operating system and user space memory"
- Begin to consider that under NGSCB even if Windows is completely compromised the NGSCB components can still provide privacy and integrity

Sealed Storage

- *This storage mechanism uses encryption to help ensure the privacy of NGSCB data that persists on the hard disk of NGSCB-capable computers*
  - provides the notion of a long-lived confidential binding between an application and its data
  - on local hard drive
  - implemented directly in hardware

Sealed Storage-Cont'd

- Sealed Storage addresses the threat that even if another operating system is booted or if the hard drive is moved to another computer the data should stay confidential
  - boot another OS
  - boot OS from CD (and chroot)
  - ignore ACLs

Attestation

- *This occurs when a piece of code digitally signs and attests to a piece of data, helping to confirm to the recipient that the data was constructed by a cryptographically identifiable software stack*
  - This is new and different
  - Application Private Networks (APNs)
  - IPSEC and VPN technologies are (mostly) layer 4. The APN is layer 7 to layer 7.

Secure Paths to User

- *By encrypting input and output, the system creates a secure path from the keyboard and mouse to trusted applications and from those applications to a region of the computer screen. These secure paths ensure that valuable information remains private and unaltered*

Secure Paths to User-Cont'd

- On keyboards each keystroke is protected (encrypted) until consumed by secure mode application
- Screen data is protected until consumed by graphic adapter, and then the window is protected from scraping

Implementation

- Bimodal
- Left Hand Side (LHS) – <u>insecure</u>
- Right Hand Side (RHS) - <u>secure</u>

Left Hand Side

- Without NGSCB HW features enabled the computer is the same as current generation PCs
- All OS's and apps should run the same
- If NGSCB is enabled and new windows OS is running then one device driver added (NexusMgr.Sys)
- Supposedly only about 200 lines of kernel code in new Windows to accommodate NGSCB

LHS Cont'd

- NexusMgr (on LHS) provides for the secure RHS:
  - All I/O (device driver access)
  - File system access
  - Memory management
  - Windows Management/Coordination
  - User Debugging?

Right Hand Side

- Secure side
- An inner security perimeter—a strong barrier to LHS
- Dependent on LHS for all interface to world

Diagram of the Four Quadrants

Left Hand Side *Standard Mode*	*Right Hand Side* *Nexus Mode*
User Apps	*Agents, TUEs, TSPs,* *NCA Runtime*
Main OS, Drivers *HAL*	*Nexus* *NAL*
Hardware	

RHS-HW Changes

- A mode flag
- Context switching
- Protect memory for RHS
- Protect RHS memory from DMA and bus mastering devices
- Load Nexus

RHS-HW Changes

- The major new component is an SSC (System Security Component)
- Also known as the TPM (Trusted Platform Module) by the Trusted Computing Group
- Provides for RSA, AES, and SHA-1 hash operations
- Contains at least one RSA private key and an AES key

RHS-HW Changes

- Intel NGSCB architecture is called *Le Grande*
- AMD presently uses the name *Secure Execution Mode*—they say 2 or 3 years away
- New keyboards
- New video cards

Nexus

Left Hand Side *Standard Mode*	*Right Hand Side* *Nexus Mode*
User Apps	*Agents, TUEs, TSPs,* *NCA Runtime*
Main OS, Drivers *HAL*	***Nexus*** *NAL*
Hardware	

Nexus

- Mini-OS, a security kernel
- Limitation of 100K to 300K lines of code for review and provability
- Will be made available for external review (terms?)
- MS will offer method to vet that source code matches shipped executable code(?)

Nexus Design

- Nexus is unconditionally trusted
- NO device drivers
- No file system, no I/O (all interface to outside world goes left)
- No paging

Nexus Design

- Nexus supports memory, process, and thread management
- Is interruptible—state is saved and control is passed left for replay

Nexus Design

- Nexus loaded by LHS
- The CPU provides an operation for an atomic hash of the nexus image at load time
- User will control what Nexus to be loaded
- Nexus can be loaded/unloaded at anytime without affecting LHS OS

Nexus Design

- LHS (the OS) can DOS the machine and/or Nexus
- But the LHS <u>cannot</u> affect confidentiality or integrity of RHS

Secure RHS Applications

Left Hand Side *Standard Mode*	*Right Hand Side* *Nexus Mode*
User Apps	***Agents***, *TUEs, TSPs,* ***NCA*** *Runtime*
Main OS, Drivers *HAL*	*Nexus* *NAL*
Hardware	

Nexus Computing Agents

- Nexus applications are known as NCAs
- User should be choosy about what NCAs to allow (only *good* ones)
- An NCA is described by a *manifest*

Nexus Computing Agents-Cont'd

- The manifest is an XML document which includes:
  - Names, version numbers, and modules to be loaded
  - Hash(es) of the module(s)
  - Public key to vet them
  - May define rules for debugging (?)

Trusted Service Providers (TSP)

- Provide common library function
- Functionally similar to DLLs but implementation different

Trusted User Interface Engine (TUE)

- XML based dialogue management (very basic dialogues in initial release—no rich dialogue on RHS)
- Also called a mini-MFC

Secure Computing Background

- James P. Anderson's "Computer Security Technology Planning Study" October, 1972
- Used extensively in "Trusted Computer System Evaluation Criteria", the *Orange Book*

Reference Monitor

- “to validate <u>all</u> references to programs or data according to the access authority of the user on whose behalf the program is executing”

Reference Monitor

- 3 operating principles:
  - the reference validation mechanism must be tamper proof
  - the reference validation mechanism must always be involved
  - the reference validation mechanism must be small enough to be tested (exhaustively if necessary)

Reference Monitor

- Concepts can be remembered via acronym *NEAT* (3 concepts become 4!)
- Non-bypassable
- Evaluate-able
- Always invoked
- Tamperproof

TCB

- Review: TCSEC defined TCB (Trusted Computing Base) as those portions of a system that are relevant to security
- Now more generally defined as all components (HW, SW, <u>and</u> procedures) that are required for enforcement of security policy

TCB and Trusted Path

- TCSEC:
  - The TCB shall support a trusted communication path between itself and users for use when a positive TCB-to-user connection is required (e.g., login, change subject security level). Communications via this trusted path shall be activated exclusively by a user of the TCB and shall be logically isolated and unmistakably distinguishable from other paths

NGSCB Limitations

- CI + AA *not* CIA + AAA
- Even if NGSCB meets its goals there will be no guaranty of availability
- System resource access (CPU scheduling and I/O) are discretion of LHS
- DOS threats may still be the same

System Audit Log, as an example

- AA *not* AAA
- Goal of audit is an immutable log (data cannot be modified, appended to, deleted, or renamed)
- Guaranty of integrity is of great importance

System Audit Log-Cont'd

- NGSCB could guaranty portions of requirement (Sealed Storage/data not changed or appended to)
- But there is no method to guaranty the data is not deleted (or even that it was ever written to disk in the first place)

Limitations-Rich Dialogue

- The presentation stack on RHS will be trim and extremely limited
- Graphic content and rich GUI will not be available (at least initially)
- Only small portions of an app may move to right (e.g. input box)
- There will be no way to sandbox a browser (again, at least initially)

Attack Vectors

- DOS is acknowledged
- Again, the LHS still caves with buffer overflows
- Also physical access and enough time means the box can be broken

BOBE

- Break Once Break Everywhere
- MS says NGSCB is not BOBE
- MS acknowledges that NGSCB SSC's private keys may be able to be pried out
- Expense/effort should be high
- MS: Only the one machine is affected

Virtualization

- If NGSCB HW can be virtualized
- Then instructions aren’t atomic and all memory can be modified
- Then pried private keys may be portable and replayable
- Then APN integrity could be spoofed...

Virtualization

- Integrity will also depend on PKI techniques
- CRLs, CDPs, and timing will be big issues—as they are today
- Changes for PKI, new requirements, and implementation/operation have not been defined

Other Limitations

- Viruses, worms may cause same amount of damage—you could still become a zombie
- Buffer overflows...

Wrap Up

- NGSCB, formerly Palladium, in conjunction with the TCG is an initiative to provide: *verifiable C + I* in terms of security, *trustworthiness* in terms of perception and marketing

Wrap Up

- The four design goals are:
  - Secure Process Isolation
  - Sealed Storage
  - Attestation
  - Secure Paths to User

Wrap Up

- HW changes include a security chip called a TPM or SSC, and CPU changes, to support the partitioning into LHS (insecure) and RHS (secure)
- SW changes include a new, small, security kernel called a Nexus to run secure applications called NCAs

Wrap Up

- The characteristics of the Nexus are similar to a Reference Monitor, which harkens back to beginnings of computer security literature
- A major attack vector is virtualization which could break the otherwise strong integrity chain
- DOS lives!

Links

- James Anderson ('72) and TCSEC: http://seclab.cs.ucdavis.edu/projects/history/seminal.html
- Microsoft: http://www.microsoft.com/ngscb
- Ross Anderson: http://www.cl.cam.ac.uk/~rja14/tcpa-faq.html
- Richard Stallman: http://www.gnu.org/philosophy/can-you-trust.html

Links

- The paper on which this presentation is based at the NASA SEWP Security Center site: http://www.sewpsc.sewp.nasa.gov

Questions?

Thank you for attending.