Migrating Application to NGSCB

Ellen Cram
Security Business Unit
Microsoft Corporation

Microsoft

Intel Developer Forum

Safer Computing Track – Fall IDF

Tuesday

- LT Overview
  SCMS-16
- TCG & TPM v1.2
  SCMS-17
- LT Architecture
  SCMS-18
- Tech Showcase
  Every Day
  Birds of a Feather Lunches
  Tuesday & Wednesday

Wednesday

- Privacy Method for Assuring Trust
  SCMS-19
- Opt-In Strategy
  SCMS-156
- Trusted Mobile KB Controller
  SCMS-24
- Software for LT
  SCMS-20
- Fundamentals for NGSCB
  SCMS-21
- **Migrating Apps to NGSCB**
  SCMS-22

Thursday

- TPM Recovery
  SCMS-25
- TCG Credentials
  SCMS-157
- TPM Mfg & Testing
  SCMS-180

= Overview
= Medium Technical
= Highly Technical

Microsoft

Intel Developer Forum

Terminology Translator

Microsoft	Intel
Left Hand Side (LHS)	Standard Partition
Right Hand Side (RHS)	Protected Partition
NGSCB Agents	RHS applications Also known as –Nexus Computing Agents (NCAs) –Agents
NGSCB Nexus	Protected OS Kernel / Domain Manager

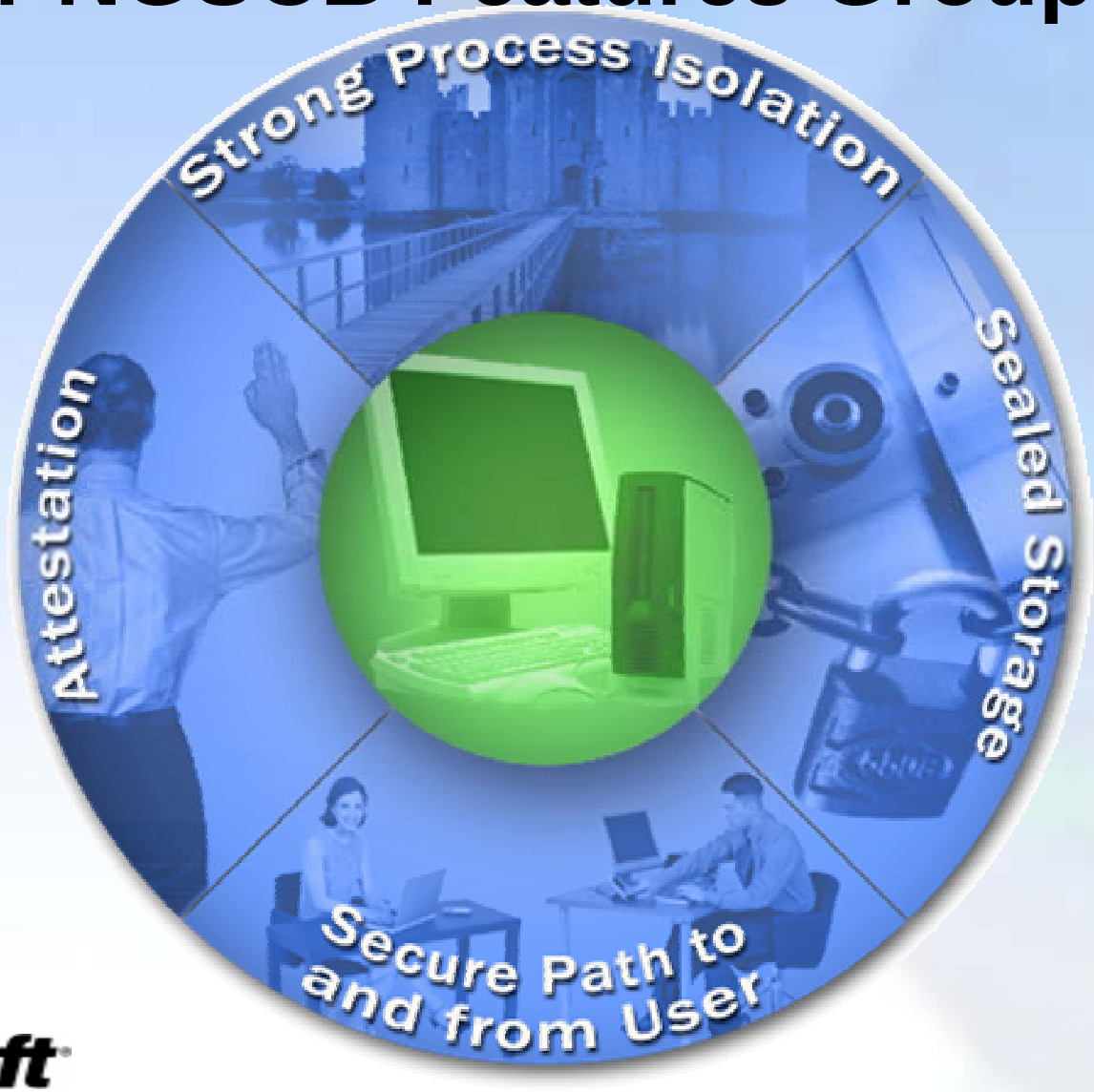
Agenda

- NGSCB Features
  - Strong process isolation
  - Sealed storage
  - Secure path
  - Attestation
- Types of NGSCB Agents
  - Application agents
  - Component agents
  - Service provider agents
- NGSCB development tools and considerations
  - Inter-Process Communication
  - Debugging
  - Manifests
  - System policy
- Summary
- Q & A

Microsoft

Intel Developer Forum

Four NGSCB Features Groups



Microsoft

Intel Developer Forum

What Does This Mean to Developers?

- All NGSCB-enabled application capabilities build off of four key features
  - Strong process isolation
  - Sealed storage
  - Secure path
  - Attestation
- The first three are needed to protect against malicious code
- Attestation breaks new ground in distributed computing
  - “Things” (software, machines, services) can be securely authenticated
  - This is separate from user authentication

Microsoft

Intel Developer Forum

Strong Process Isolation

- Agents run in curtained memory
  - Not accessible by the standard Windows kernel
  - Not accessible by hardware DMA
  - Not accessible by other agents
- Enforced by LT hardware and NGSCB software
  - Nexus arbitrates page tables
- Process isolation is inherent in NGSCB
  - No work needed to utilize it
- All communication between agents must be done through an IPC layer
  - Memory cannot be shared in NGSCB

Secure Path To User

- Secure input
  - Secure session between device and nexus
  - Protects both keyboard and mouse
- Secure output
  - Secure channel between graphics adaptor and nexus
- Secure I/O is inherent in NGSCB
- Trusted User Engine (TUE) offers higher-level services for agent developers
  - Window layout is defined by the agent using an XML based format
  - Window interaction is managed by TUE
  - Events are passed to the agent

Microsoft

Intel Developer Forum

Using The TUE – Code Snippet

```
CONST WCHAR *gHelloWorld =
L"<DockPanel>"
L"    <FlowPanel Margin=\"5\" Dock=\"Top\">Hello World!</FlowPanel>"
L"    <FlowPanel Dock=\"Top\" ID=\"ButtonPanel\">"
L"        <Button ID=\"Add\" Margin=\"3\">Add</Button>"
L"    </FlowPanel>"
L"</DockPanel>";

CONST WCHAR *gOkButton = L"<Button ID=\"Ok\" Margin=\"3\">Ok</Button>";
static STATUS

lOkButton (    IN NX_UI_HANDLE Dlg, IN NX_UI_HANDLE Dispatch,  IN CONST WCHAR *ID, IN VOID *Data )
/* Routine Description: This causes the UI to close by signalling the "done" parameter passed into this through the "Data" parameter.*/
{
    PBOOL done = (PBOOL)Data;
    *done = TRUE;
    return STATUS_SUCCESS;
}

static STATUS

lAddButton ( IN NX_UI_HANDLE Dlg, IN NX_UI_HANDLE Dispatch, IN CONST WCHAR *ID, IN VOID *Data )

/* Routine Description:   This replaces the contents of the element ButtonPanel with the XML fragment    gOkButton. visually this replaces the button "Add"
        with the button "Ok".  An event handler for the OK button is then installed */
{
    STATUS status;

    status = NxRendSetXMLContent( Dlg, L"ButtonPanel", gOkButton);

    if (SUCCESS( status )) {

        status = NxRendRegisterEvent( Dispatch, L"Ok", lOkButton );

        if (SUCCESS( status )) {

            status = NxRendDraw( Dlg, TRUE );
        }
    }
    return status;
```

Microsoft

Intel Developer Forum

Sealed Storage

- Provides a method for encrypting data with a key rooted in the hardware
  - Each nexus generates a random keyset on first load
  - TPM chip on motherboard protects the nexus keyset
  - Agents use nexus facilities to seal (encrypt and sign) private data
  - The nexus protects the key from any other agent/application, and the hardware prevents any other nexus from gaining access to the key.
- Key ring component provides higher-level services for agent developers
  - Provides key generation, encryption and decryption APIs
  - Manages key storage, backup, and migration

Microsoft

Intel Developer Forum

Attestation

- When requested, the nexus can prepare a chain that authenticates:
  - Agent by digest, signed by the nexus
  - Nexus by digest, signed by the TPM
  - TPM by public key, signed by OEM or IT department
- Other forms of attestation are possible that provide less information
  - Using a trusted third party
  - Using a zero-knowledge proof
- The machine owner sets policy to control which forms of attestation each NCA or group of NCAs can use
- Secure communications agent provides higher-level services to agent developers
  - Open a secure channel to a service using a secure session key
  - Respond to an attestation challenge from the service based on user policy

Types Of Agents

- **Application agents are intended to run as stand-alone applications.**
  - **The entire application runs on the RHS.**
- **Component agents are designed to appear to a standard application as an external COM or managed object**
  - **Most of the app runs on the LHS and agents are used for specific trusted operations**
  - **A LHS proxy translates between COM or .Net and NGSCB IPC**
- **Service provider agents (SPAs) provide services to other agents**
  - **Requests and responses go over IPC**
  - **SPAs are not available to LHS applications**

Agent Considerations

- Most applications will be either Application or Component agents, depending on the problem to be solved
- Application agents are good for clients in multi-tier applications
- Component agents are good for adding trusted features to existing Windows apps
  - Component agents look like a COM or managed object to an LHS application
    - There is an actual COM or managed object that proxies requests over to the agent
    - NGSCB will include a tool to generate proxies and stubs based on IDL
  - Component agents are an easy way to extend Web sites and existing applications
- Agents are monolithic - no DLLs
  - Code can be shared using statically-linked libraries
  - Composition of agents is based on IPC

Sample Component Agents

- Digital signature agent
  - Takes a pre-rendered (WVG) document, displays it to the user, and securely signs it
  - Ensures “what you see is what you sign”
  - Provides secure private key storage
- Confidential data viewer
  - Takes a URL to a Web service, fetches XML-formatted data, and displays it to the user
  - Secures the data against screen scraping viruses
  - If interactive, protects entered data against keystroke sniffers

Standard Service Provider Agents

- NGSCB includes several standard SPAs
  - Some of these are exposed through the nexus API, rather than by explicit IPC
- The Trusted UI Engine (TUE) provides a dialog, form-oriented user interface based on an XML dialog definition language
- The Policy Engine provides a general-purpose authorization mechanism
  - XrML 2 is used as the policy expression language
- The Key Ring provides key generation, encryption and decryption APIs, and manages key storage, backup, and migration
- The Secure Communications Agent provides a general mechanism for creating a secure, attested networking channel to a Web service

Inter-Process Communication

- IPC is asynchronous and message-oriented
- Agents and LHS processes can both use IPC
  - Agents can communicate with other agents
  - LHS applications can communicate with agents they start
- Access to IPC is controlled by policy

Writing NGSCB Agents

- Agents may be written in C or C++, using any compiler
- Once we have a RHS CLR, agents will be able to be written in any .Net language
  - The RHS CLR is planned to ship shortly after RTM
- Two classes of RHS functions:
  - Functions enhanced by NGSCB
    - We use the hardware and protected environment to make these calls safer than they would be on standard Windows
  - Functions not protected by NGSCB
    - Indicated by a specific prefix
    - These functions are not any safer than an equivalent function in standard Windows
    - Our goal is to enhance these functions with NGSCB in future versions

User Mode Debugging

- Agents are only debuggable if set in the manifest
  - Changing the manifest to enable debugging changes the agent's code identity
  - This change is reflected by attestation
- Debugging an agent really means debugging a LHS shadow process
  - We've redirected the functions to Get and Set Thread Context and Read and Write Process Memory
  - We've redirected RHS debug events to the LHS process
  - Thread control "just works"
- All well behaved debuggers that work with LHS processes will also with agents

Debugging The Nexus

- The retail nexus cannot be debugged
- We will provide a debug version of the nexus to enable system debugging
- Since these two nexuses are different in at least one bit, their attestations are different as well
- We support the same debugging technology as in the LHS kernel into the debug nexus

Agent Manifest

- Signed XML document that defines:
  - Agent components
  - Agent properties
  - Agent policy requests
  - XML schema is an NGSCB-specific extension to the standard Longhorn manifest
- Policy requests are not binding
  - Machine owner policy overrides manifest policy requests

Manifest Details

- Agent properties
  - System requirements
    - E.g. Debugable = FALSE
    - Enforced by NGSCB
  - Descriptive properties
    - E.g. Version = 1.1.2.2
    - Not interpreted nor enforced by the system
  - Secret migration defaults
- Requested rights/privileges
  - E.g. access to trusted output, write access to a counter, etc.

System Policy

- Set by the machine owner
  - The owner may allow users the ability to override or extend
  - The owner may choose to delegate policy & trust decisions to a 3rd party
    - “Use EFF’s policies for any agent signed by Foo”
    - “Use my IT department’s policies for all agents”
- Expressed using signed XrML policy certificates
- Policy is checked at run-time for every request
  - Some policy decisions are cached in the nexus for performance reasons

Resources Controlled By Policy

- Running an agent
- Responding to an attestation challenge
  - Responding to an attestation challenge with full disclosure
- Accessing a specific secret
- Accessing the sealed storage API set
- Accessing the network API set
- Accessing the file system API set
- Creating a child process
- Accessing the TUE
- Lots more…

Summary

- NGSCB is made up of four key features
  - Strong process isolation
  - Sealed storage
  - Secure path
  - Attestation
- These features can be utilized either through stand-alone or componentized applications
- The NGSCB development environment is similar, although more constrained, than the standard Windows environment

Microsoft

Intel Developer Forum

Next Steps

- Study the code
- Come to Microsoft Professional Developers Conference (PDC)
  - Developer Preview SDK and Tools
  - October 26-30, 2003 in Los Angeles
  - http://msdn.microsoft.com/events/pdc/
- Ask your vendors what NGSCB-enabled components they will provide
- Read the available white papers and specs

Microsoft

Intel Developer Forum

Resources

- Visit our site
  - http://www.microsoft.com/ngscb
- Q&A
  - ngscb_qa@microsoft.com
- E-Mail updates
  - Subscribe to the WTPI information newsletter for ongoing updates; send blank e-mail to
  - wtpiinfo-subscribe@pens.tm500.com

Microsoft

Intel Developer Forum

Fall ’03 U.S. IDF session presentations are available to IDF attendees only. To download, go to:

`http://www.intel.com/idf/attendee`

`Username:  attendee`
`Password:  fall2003`

Microsoft

Intel Developer Forum

Thank you for attending.

Please fill out the Session Evaluation Form.

Microsoft

Intel Developer Forum