Trustworthy Platforms

Problems, Promises, Concepts, Practical Realities, and Research Opportunities

George Cox

Principal Security Architect
Intel Corporation

04/28/2005

Problems

- Our computing platforms, from cell phones to servers, find themselves under ever more aggressive and tenacious attack
- As we continue to become more dependent on these platforms in our daily lives, we naturally want to be able to trust that they will not fail under these attacks
- Building a basis for being able to trust the robustness of our computing platforms is the subject of this talk

We Are Here

- Market expectations for product security and robustness are changing:
  - From being a necessary evil
  - Through being a valuable advantage
  - To being a required checklist item
- But, more work is required because of problems like:
  - Digital identity theft and consequent fraudulent use growing rampant
  - Unauthenticated/unauthorized digital actions commonplace
  - Unauthorized code sneaking onto and running on your platforms

Security and robustness are not absolutes – rather, they require continual "bar raising"

Definitions

- Security functionality
  - Requirements – Market expectations, standards, profiles
  - Solutions
    - Surface/visible functionality
    - Specified, implemented, completed, validated, interoperable, certified, delivered, …
- Secure/robust (security or other) functionality
  - Robust against bugs or attacks
  - Achieving
    - Reduced attackability of feature implementations and runtime environment
    - Via improved integrity, privacy, authentication/authorization, …
    - Through use of a variety of security technologies

Trustworthy Platforms – set of platform elements and services available to help provide "secure/robust" functionality

Promises

- When we can do the "heavy lifting" to get broad integration and deployment of deploy
  - Secure, robust, trustworthy platforms (for clients, servers, handhelds, …)
  - And the necessary ecosystem infrastructure to provide
    - Strong authentication and authorization infrastructure (HCI and CtoC) and
    - Tightly controlled provisioning infrastructure (strong module management lifetime model)
- We can potentially make…
  - Digital identify theft and use a thing of the past
  - Unauthenticated/unauthorized digital actions a thing of the past
  - Unauthorized code a thing of the past

Concepts

- Secure communication between communicating agents
  - Communication protocol stacks @ each end provide
    - Authentication
    - Message integrity
    - Message privacy
    - Authorization/access control
- Secure the environments/platforms in which the communicating agents run
  - Design of the environments/platforms provide
    - Enforced separation between elements
    - Minimum TCB of elements
    - Authentication, authorization, and access control between elements
    - Measured and attestable integrity of elements
    - Measured and attestable installation/replacement of elements
    - Attestation-based trust generation between communicating agents

Trust Generation

- At least, be able to tell the party you want to trust you about
  - Your configuration
    - => the configuration is known
      - Including all the HW, SW, and data elements that statically comprise the configuration and
      - A protected record of
        - The current static configuration (e.g., signed hashes per element);
        - The steps that it went through to get there; and
        - The steps that the current dynamic configuration has gone through since startup
    - and
    - => can be "measured" to see if the current configuration matches what the protected record says that it should be
    - => protected mechanism for inspecting/hashing all necessary elements of the local configuration
    - => basis for erroneous configuration detection and recovery
  - Projected across space and time
  - In a cryptographically unforgeable way
    - => protected local unique keying material for signing
    - => protected signing functionality (hashing and asymmetric encryption/decryption)

A Trustworthy Platform

- Provides systemic answers to model implementation issues
  - Separation – processes, VMs, multi-core, OOB capabilities
- Using the best currently available widely reviewed runtime base
  - Policy-based MAC for "kernel" services
  - Such as facilities evolving in Linux/SELinux and "trusted" Xen
- Depending upon a HW-based "root of trust" - TPMish semantics
- With major steps forward in system/security structure
  - Trusted boot => through TCB and applications
  - Partitioned communication stacks
- With additions to address service issues
  - Equivalent MAC for local and remote "non-kernel" services
  - Such as facilities evolving in SELinux User Space Object Managers
- With appropriate privacy protection approaches
- Providing the basis for certification and standards
  - Common Criteria EAL 4 M certified foundation
  - WS* security for WS-Management
- While maintaining existing external and internal communication and programming interfaces

Practical Deployment Problems

- Associated infrastructure tools/SDKs
  - Authentication/authorization tools
    - Credential manufacturing, signing, provisioning
  - Module management tools
    - Manifest manufacturing, signing, provisioning
  - Policy tools
    - Policy manufacturing, signing, provisioning

Research/Collaboration Opportunities

- Integrity enforcement
  - Integrity Measured Linux - tcgLinux –Van Doorn/Sailer et al
  - Attestation-based Policy Enforcement for Remote Access – Van Doorn/Sailer et al
  - Minimal Integrity Protected TCB – Sailer/Jaeger et al
  - Comparable work by Johns Hopkins/Mitre
- Attestation models/protocols - what to "attest" to whom
- TPM evolution
  - Multiple/virtual contexts – multiple domains/owners per platform
  - Active TPMs – slave coprocessor => peer coprocessor
  - Script driven TPMs – evolutionary additions (even to legacy platforms)
- Infrastructure
  - TCG provisioning work
  - Web Services-Management work