Recollections about
Data Integrity Limitations in
Hybrid Security Architectures

Workshop on Innovations in Strong Access Control
Monterey, CA
September 25-27, 2000
Cynthia Irvine and Timothy Levin
Center for INFOSEC Studies and Research
Computer Science Department
Naval Postgraduate School, Monterey, California, USA
irvine(levin)@cs.nps.navy.mil

Sponsorship and Related Work

• This work was sponsored in part by:
  – the Community Intelligence Office and
  – the DARPA/ITO Quorum program

• Principal aspects of this presentation were recorded as a technical report in:
Outline

- Background
  - Integrity
  - High Assurance Systems
- Hybrid Security Architectures
- Integrity and HSAs
- Conclusion

Integrity

- Dual of Confidentiality
  - Labels indicate potential loss from
    - unauthorized modification
    - vs. unauthorized disclosure
- Translation from Confidentiality
  - inverted/convoluted
  - difficult concepts [Gasser]
- Analysis can be overlooked
Integrity (2)

• Inherent Integrity and Confidentiality
  – explicit labels
  – or implicitly understood
• Unclear distinctions/assumptions
  – High confidentiality implies high integrity?
• Integrity of Code
  – fidelity to original - e.g., distributed version
  – fidelity to described intent
    • correct functionality
  – no additional functionality
    • trap doors, Trojan horses

High Assurance Systems

• Enforce confidentiality and integrity
  – to defined degree of assurance
• Various architectural approaches...
In the Beginning...

Distributed
Vertically Distributed

- User
  - Application
  - ORB
  - RDBMS
  - File Server

Need to Generalize Remote Access

- remote programs
- remote data
- remote processors, devices
High Assurance Systems

- Expensive
- Incompatible, “stovepipes”
- Responses
  - COTS in Government RFPs
  - Balanced Assurance at vendor initiative
Balanced Assurance

- Not Trusted COTS Application
- RDBMS
- EAL5 DAC
- Trusted B3 MAC

Hybrid Security Architecture

- Balanced Assurance + COTS + MLS
- Configuration Components
  - Untrusted COTS terminals/workstations
  - Untrusted COTS applications
  - Storage devices
    - Multilevel data
  - Multilevel TCB mechanisms (RVM)
  - TCB extensions
  - Network Connections
    - Single and multilevel
### Monolithic

<table>
<thead>
<tr>
<th>High COTS User Interface</th>
<th>Low COTS User Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>COTS Application</td>
<td>COTS Application</td>
</tr>
<tr>
<td>Reference Validation Mechanism</td>
<td></td>
</tr>
</tbody>
</table>

- **Not Trusted**
- **Trusted**

### Switch-Based

- COTS Client User Interface
- RVM

- **Not Trusted**
- **Trusted**
Distributed

MLS Access Enforcement

• Reference Validation Mechanism
  – mediates access to objects
  – controls object creation, storage, access, I/O
  – prevents data leakage across MAC partitions
    • confidentiality write-down or read-up
    • integrity read-down or write-up (if enforced)

• Other modules constrained by RVM
  – leakage
System Confidentiality Capacity

• System trusted for confidentiality
  – to confidentiality limit of reference validation module
    • Assigned range
    • Yellow Book
      – Maps assurance levels to confidentiality levels

Data Integrity

• Integrity Semantics
  – dual of confidentiality
    • “prevents data contamination from untrusted software”
  – was the modification correct?
    • Within the partition
    • Code trusted to handle data correctly
      – to its level of assurance
    • No Yellow Book for integrity
    • Look to code integrity label
Code Module Integrity Label

- What the system designer needs it to be
  - Coherent network architecture
  - Limit: pedigree of code

Trust in Low-Assurance Code

- Evaluation below B2/EAL5
  - little or no code review required
    - no examination for Trojan horses/trap doors
    - no code correspondence
  - no trusted distribution
  - potential for unknown functionality
    - e.g., “Easter eggs” common in commercial software
    - testing doesn’t address unknowns
- Integrity untrusted (“low assurance”)
  - integrity label
System Integrity Capacity

• *Data-Path Modules* handle data between user and store
  • Without integrity enforcement all DPM could modify data (*Data-Modifying Path Modules*)
  • Integrity enforcement can limit which modules can be DMPM
    – depending on module and data labels

• System trusted for integrity
  • to integrity limit of least-trusted DMPM
    – with or without RVM integrity enforcement

Integrity Capacity

• Two Cases
  – Integrity not supported
    • system can take in data higher in integrity than system
    • data output is lowered to integrity of code
      – de facto label
  – Integrity Supported by RVM
    • cannot take in data higher in integrity than code
      – problem not addressed by ring mechanisms
    • system regulates its own integrity capacity
    • data is not corrupted by less-trusted code
HSA and Integrity

- HSA applications and user interfaces
  - COTS
  - below B2/EAL5 (integrity untrusted)
  - generally designed to modify data
- HSA systems have untrusted or low assurance integrity capacity
- Hybrid Security Architecture systems not suitable for AIS environments with critical data integrity requirements

Summary

- *Code-module integrity* limits *system integrity capacity*
  - Not new information
  - Not always remembered
  - Not always communicated to sponsors and customers
- HSA systems not suitable for environments that have critical data integrity requirements