

# The Challenge of Multilevel Security

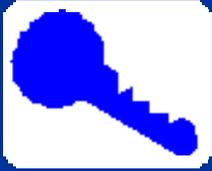


Rick Smith, Ph.D., CISSP

Rick@cryptosmith.com

<http://www.cryptosmith.com/>

October 2003



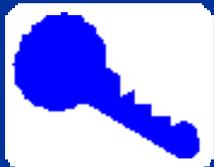
## **Text-Only Outline**

*Outline presented here*

- **What is MLS?**
- **Why is MLS Hard? – Accreditation**
- **Building MLS Systems**
- **Selecting a Trusted OS**

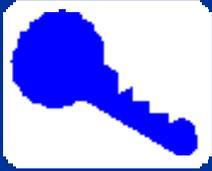
*Please see the BlackHat CDRom for the complete copy of this presentation, or visit this web site:*

<http://www.cryptosmith.com>

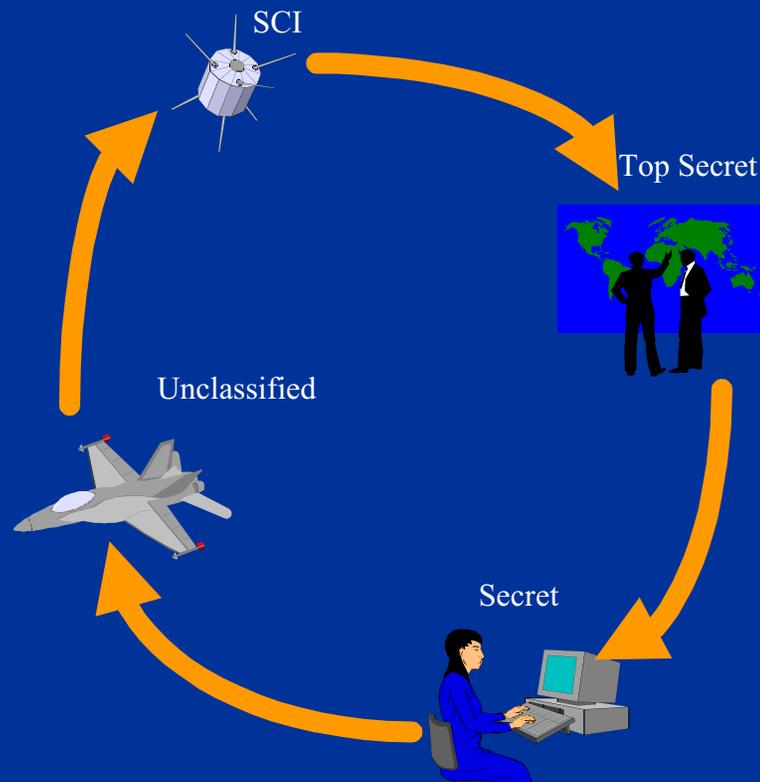


# Multilevel Security

- An overloaded term
- Some vendors build “MLS Products”
  - Implement “Bell LaPadula” security mechanism
  - Allows higher-classified processes to read data created by lower-classified processes
  - Example: a Top Secret user’s process can read Secret data
  - Vice versa (downgrading) not directly permitted
- Most requirements for “MLS Operating Mode”
  - Devices handle classified information with different classification markings
  - Must never release wrong level to wrong recipient
  - Much more general than “MLS Products”

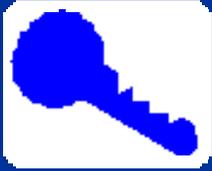


# An Example MLS Problem



**Sensor to Shooter:**  
Data travels from satellites to planners at different levels, and finally to the warrior who pulls the trigger.

Data is sanitized at each level and passed to a lower classification.



## **MILS versus MLS**

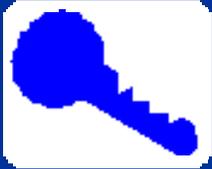
*Achieves “MLS Operating Mode”  
without “MLS Products”*

- **MILS = Multiple Independent Levels of Security**
  - Deals with multiple levels via separate, “System High” elements
  - Data sharing, if any, is via guards or one-way data transfers
- **Does not necessarily require “MLS Products”**
  - Most or all elements may be standard COTS products
  - Guard may use an MLS Product, but not necessarily
- **Site networks usually operate in “MILS” mode**
  - Individual networks consist of COTS products
  - Networks run at System High
  - Interconnections, if any, require a special-purpose Guard



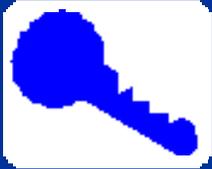
## Why is MLS Hard?

- **Short answer: Software is unreliable**
  - Nobody wants to trust the protection of their own, valuable classified information to a buggy OS or application
  - *Felony Boxes* – nobody wants to be personally liable for leaking classified information
- **MLS accreditation tries to reduce/eliminate risk**
  - Accreditation – approval to operate by major command user
  - MLS accreditation seeks to eliminate risk of data leaks
  - Confidence in software = confidence in safety of data
- **Modern software is too complex for confidence**
  - 16 million lines of code in modern Windows OS



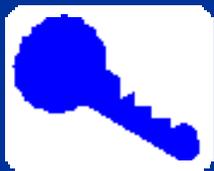
# System Accreditation

- **Required of all systems handling classified data**
- **Regulations: DOD 5200.1, now DOD 8500**
  - Regulations establishing policies for DOD info systems
- **DITSCAP: Defense Information Technology Security Certification and Accreditation Process**
  - Process to verify a system's security features – “certification”
  - Process to authorize its operation – “accreditation”
- **SSAA – System Security Authorization Agreement**
  - Documents security requirements, features, and steps taken to assure its correct and secure operation
- **DAA – Designated Approval Authority**
  - General/Flag officer at major command
  - Signs of on need and risk for using the accredited system



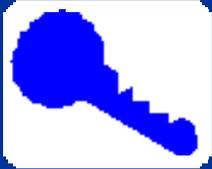
# Getting Into Operation

- **“Full” Accreditation**
  - System goes through certification process
    - May be based on evaluations of products being used
    - May be based on template of another successful site – this is how the SABI/TSABI processes work
    - May involve a combination
  - DAA approves system for operation
- **IATO – Interim Approval to Operate**
  - Certification is incomplete; DAA lacks basis to fully accredit
  - May occur in “emergency” situations where system is needed regardless of the certification status and risks
  - At the discretion of the major command’s DAA
  - DAA may even make an IATO permanent (“back door” approval)



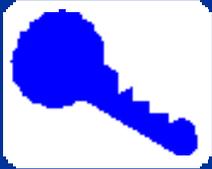
## Evaluation: a product-oriented process

- **Process established by data owner(s)**
  - Pioneered by NSA: Owner/producer of classified information
  - Evaluated systems to serve as surrogates to enforce NSA policy
- **Expects vendors to seek product evaluation**
  - Historically, this is the exception, not the rule
- **Evaluation is supposed to “authorize” use**
  - Traditionally, MLS systems had to achieve a certain level of evaluation and incorporate certain features: “B1” or “EAL4”
  - In practice, the DAA is the final authority
- **In practice, evaluation becomes one more factor**
  - Some MLS systems use evaluated products
  - Some MLS systems rely on other assurances



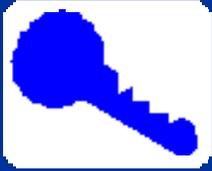
# SABI/TSABI

- **(T)SABI = (Top) Secret And Below Interoperability**
- **Process established by end users**
  - Pioneered by the ASD/C3I and the JCS
  - Representing warfighters, not data producers
- **Focus on guards connecting MILS networks**
  - Particularly DISA and NSA networks
- **End user initiates the process**
  - posts a “ticket” defining what they need to do
  - SABI/TSABI provides templates for common guard configs
  - New solutions may serve as templates for future users



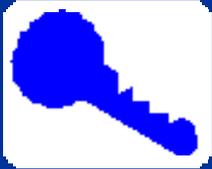
# Program Risk

- **No process guarantees accreditation**
- **Evaluations, SABI, TSABI, etc., try to reduce risk**
  - Provides evidence of correctness to help convince accreditors
  - Policy or prior accreditations used to support arguments
- **Assurance vs Cost Trade-off**
  - Evaluations, SABI, TSABI processes increase assurance
  - High assurance increases product costs
  - Cheaper, COTS products provide lower assurance



# Building MLS Systems

- **Establish the networking infrastructure**
  - Option: physical separation
  - Option: system-high LANs with separation
  - Option: MLS LANs with Type 1 encryption
- **Establish low-to-high flows**
  - One-way optical transmission
  - MLS middleware with read-down capabilities
- **Establish high-to-low flows - downgraders**
  - Manual review on COTS platforms
  - Manual review on a trusted platform
  - Automatic review/sanitization by a trusted guard



# Network Infrastructure

- **Wiring has its own problems**
  - Physical protection, separation, auditing, assurance
- **System-high LANs**
  - Provide separation, not confidentiality
  - Examples: Dragonfly, Cryptek's DiamondTEK
  - Issue: must physically protect confidentiality of LAN
- **Network encryption minimizes wiring**
  - Confidentiality using Type 1 encryption
  - Examples: GD Fastlane/Taclane
  - Share internal LAN wiring to minimize extra wires
  - Issue: infrastructure costs of Type 1 encryption

**GENERAL DYNAMICS**  
Advanced Information Systems

**CRYPTEK**<sup>™</sup>



# Low-High Data Flow



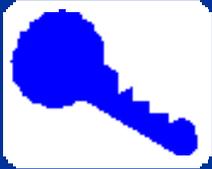
- **Option: Use one-way flow hardware**
  - Examples: Tenix, Owl
  - Ensures one-way data transfer, no backward leakage
- **Option: use guards for low-high flow**
  - Downgraders can also move data low-to-high
  - (see later discussion)
- **Option: Use middleware...**



# Middleware for Low-High Sharing

- Use approved middleware to store shared data
  - Option: multilevel web server
    - Example: TSL Trusted Web Server, TCS MLS Web Server
  - Option: multilevel database
    - Example: Trusted Oracle, Rubix
  - Option: multilevel file sharing
    - Example: TCS Trusted Gateway System
- Gap: these are *moderate assurance* solutions
  - Can not share data across a broad classification range
  - Often restricted to two adjacent classification levels
  - Broader ranges require additional network security mechanisms

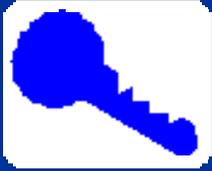




# High-to-Low Reclassification

- **Manual review for downgrading**
  - People examine and sanitize interactively
  - Option: On-the-spot reviewing on user desktop workstations
  - Option: Trusted review terminal for a disclosure officer or clerk
- **Automatic review for downgrading**
  - Mechanized rules for passing data safely
  - Issue: not all reviews can be automated effectively
- **Guards filter/sanitize the actual transfers**
  - Existing guard products: Radiant Mercury, Digitalnet SAGE, ISSE
  - Gap: some applications need custom guard filtering
    - Option: build atop existing guard
    - Option: create new guard software if existing guards inadequate

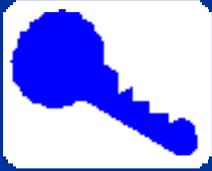




# High-to-Low Downgrading

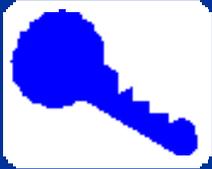
- **Option: Use OS to host a custom guard**
  - Examples: XTS-400, Aesec, Sun Trusted Solaris, SGI Trusted Irix, Green Hill Integrity 178B, Lynuxworks LynxDO178B.
- **Option: Use existing guards to filter/sanitize traffic**
  - Examples: SAGE, Radiant Mercury, ISSE Guard
- **The Gaps**
  - Must implement multilevel applications and earn accreditation
  - Need customer approval on strategy and classification filtering



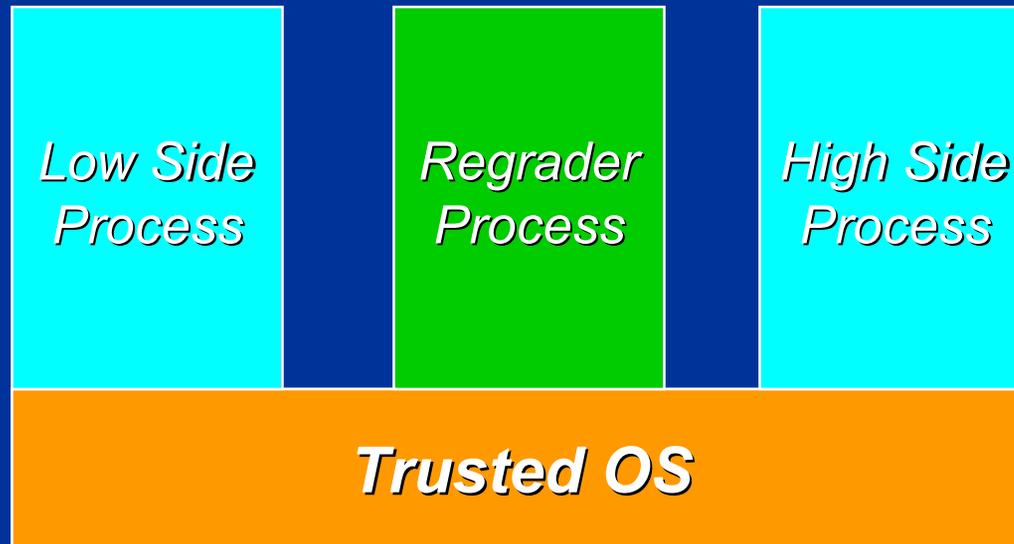


# Trusted Systems: Build vs Buy

- **Trusted software is very costly to develop**
  - Developers placed under intense scrutiny
  - Detailed documentation of software architecture, design
  - BUT – third parties charge a fortune to do this work for you
- **May be feasible to build small-scale products**
  - Small, simple software components
  - Must reside atop a trustworthy OS
- **Traditional Trusted OS Options**
  - OS with Strong Labeling
    - Examples today: Digitalnet XTS-400, Aesec Platform
  - OS with “Sufficient” Labeling
    - Examples Today: Sun Trusted Solaris, SGI Trusted Irix



## How do we use a Trusted OS?



- OS ensures process separation
- Certifiers look at processes independently
- Assured separation = easier certification

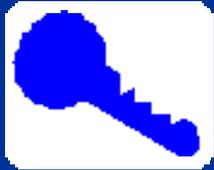
*Process separation is the key to certification*



# Emerging OS Options: Open Source

- Offer MLS and other schemes to ensure security
  - Provides the expected MLS mechanism for process separation
  - Option to use Biba or other separation mechanisms
  - Process separation is the key, not just MLS
- Example Products
  - NSA's Security Enhanced Linux (SELinux)
    - Rumor – actually been used in operational systems
  - FreeBSD with security extensions like MLS: "Trusted BSD"
- Gap: Open source lacks vendor control
  - Existing documents don't necessarily match the code
  - No assurance regarding authorship of the code

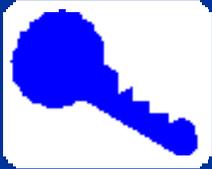




# Emerging OS Options: Safety Certified OS

- OSeS that earned highest safety certification for flight software: RTCA/DO-178B Level A.
  - RTCA: formerly “Radio Technical Commission for Aeronautics”
- Provides high assurance of process separation
  - In flight safety, ensures that a software glitch in one process won’t interfere with a different, critical software process
  - Simplifies assurance by allowing software partitioning
- Example Products
  - Green Hills DO-178B product
  - LynuxWorks – LynxOS-178 provides DO-178 assurance documents
- Gap: DO178-B doesn’t cover all security bases
  - DO178 Level A exceeds many security requirements, but
  - DO178 lacks assurances against malicious software, developers
  - Green Hills working on Common Criteria security evaluation with LM





# What About Microsoft Windows?

- Microsoft quietly speaking of MLS support
- Current direction based on NSA's NetTop work
  - Use PC-based virtual machines for level separation
    - Each "Level" has its own Windows OS
  - Separation kernel approach instead of true MLS
    - Data sharing via external mechanisms
  - Product: VMWare
- Issue: this is exploratory work
  - Microsoft has backed away from MLS support before
  - VMWare itself lacks the assurance needed for accreditation



# Thank You!



Questions? Comments?

My e-mail:

[rick@cryptosmith.com](mailto:rick@cryptosmith.com)

<http://www.cryptosmith.com>