

Real Time Security

Joint Tactical Radio Example

The Open Group

July 18, 2001

Agenda

- The Open Group Real Time Forum
- Joint Tactical (Software) Radio (JTR)
- JTR Security Architecture
- Future Directions

RT Security Forum

PURPOSE

Secure systems and real time, embedded systems are subject areas with many requirements, guidelines, and standards. Historically, real time systems designers have not included information security/assurance in their requirements. Our interest area is the intersection of real time and security. A growing number of systems exist today that have both real time (including embedded) and security considerations. The Real Time Security Forum expects the number of such systems to grow rapidly as we enter the information age of mobile, wireless connectivity.

We recognize that different system domains have different security and real time requirements, so a single solution is not the answer. There is a need for an approach that system architects can use and apply within their own system domain, and yet maintain certain standard to allow portability among these different systems. There is also a need for tools the system software developer can use to address security issues for real time systems.

RT Security Forum

SCOPE

We seek practical solutions and tools that will allow the developer to apply security policy in a flexible cost effective manner. We seek technologies and practices that can enable a wide variety of system architects to make informed decisions. As with any architecture, this involves tradeoffs by the system architect. Areas of tradeoff include:

Resource Utilization

Performance Responsiveness

Availability of Service / Robustness

Security features to support application

Cost

RT Security Forum

BACKGROUND

Real Time (RT) system requirements are currently being found in systems ranging from manufacturing to state-of-the-art tactical systems. However, the inclusion of RT systems into these processes significantly alters the overall security environment. Information security/assurance usually has not been addressed as a design constraint. Addressing those security issues will present some serious challenges to overcome. The military for example places a tremendous reliance on global information transfer to all command and fighting elements of a war. Commercial enterprises have significant information security needs in the current global environment. Such reliance demands security of this information far beyond what is present today. Not only is detection of any intrusion on this global communication important but real time detection and immediate severing of the intrusion; and continued, unhindered, operation is a strong requirement. With Real Time required for Command and Control (C2), the security safeguard of RT systems becomes extremely critical. The impact of RT on C2 has security implications for the entire infrastructure in the US and globally for all countries.

RT Security Forum

PROBLEM STATEMENT

The problem focus is upon embedded systems since they present the most difficult application area. Many embedded RT processors control critical systems. The range of critical RT systems includes chemical processes, power plants and grids, traffic control, manufacturing production lines, avionics, missiles, strategic defense systems, etc. Human lives, regulatory compliance, mission success, and even national security often depend on these RT systems properly functioning as designed and with little margin for error. Because of the unique nature of the RT operating environment, the protection of embedded RT systems presents a unique challenge for the system designer.

RT Security Forum

POSSIBLE SOLUTION

We propose as a possible solution the creation of a RT Security API. The RT/OS vendors would be asked to support a standard API so that developers could tailor the security environment to the requirements of the domains supported by the software under development. Guidelines would be developed to provide the basis for the API creation using the Open Group as the coordinating body.

These guidelines contain two components:

1. **The Security Requirements Matrix for the RT Security API and the hooks needed to implement the API in an RT/OS. This requirements matrix constitutes the bottom-line requirements/guidelines for the RT software developer and the RT/OS vendor. It would include control methods the required interrupt priorities would need for authentication, authorization, and encryption, as required.**
2. **An explanation of the overall environment of threats, vulnerabilities, and the unique security issues pertaining to the RT processing environment. This supplies the system architects with the framework for RT system design.**

Security Policy Concerns and Requirements

- Security policy concerns
 - Data isolation
 - Information flow
 - Can be extended to time isolation
- Security policy enforcement mechanism requirements
 - Non-bypassable
 - Always invoked
 - Tamper proof
 - Ability to Evaluate

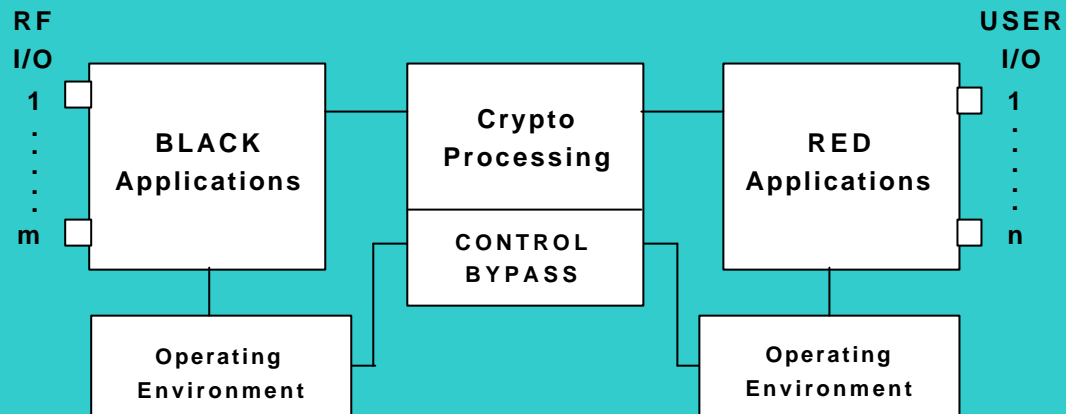


Joint Tactical Radio

- **Software Communications Architecture
Computer Security**
- *Defense-In-Depth in an Embedded
Environment*
- Abstracted from brief created by Mike Weller, Rockwell Collins, (mkweller@collins.rockwell.com) and W. Mark Vanfleet, NSA C12

JTR

Security Architecture



BLACK Side Functions

Required

- TRANSEC Stream
- TRANSEC Processing
- Access Control
- Virus Check
- HMI
- Remote Control Processing
- Authentication
- Software Integrity

Implied

- Pattern Recognition (e.g., Message Headers)
- Protected Storage

Crypto Functions

Required

- Encrypt
- Decrypt
- TRANSEC Stream
- Key Load
- Key Manage
- Algorithm Processing
- Bypass
 - User Information
 - Radio Control
- Integrity
- Authentication

Implied

- RED/BLACK Isolation (Electrical)
- RED/RED Interface
- BLK/BLK Interface

RED Side Functions

Required

- Classified Applications
- Access Control
- Identification
- Authentication
- Integrity
- Audit
- Virus Check
- Crossbanding
- Remote Control Processing
- HMI

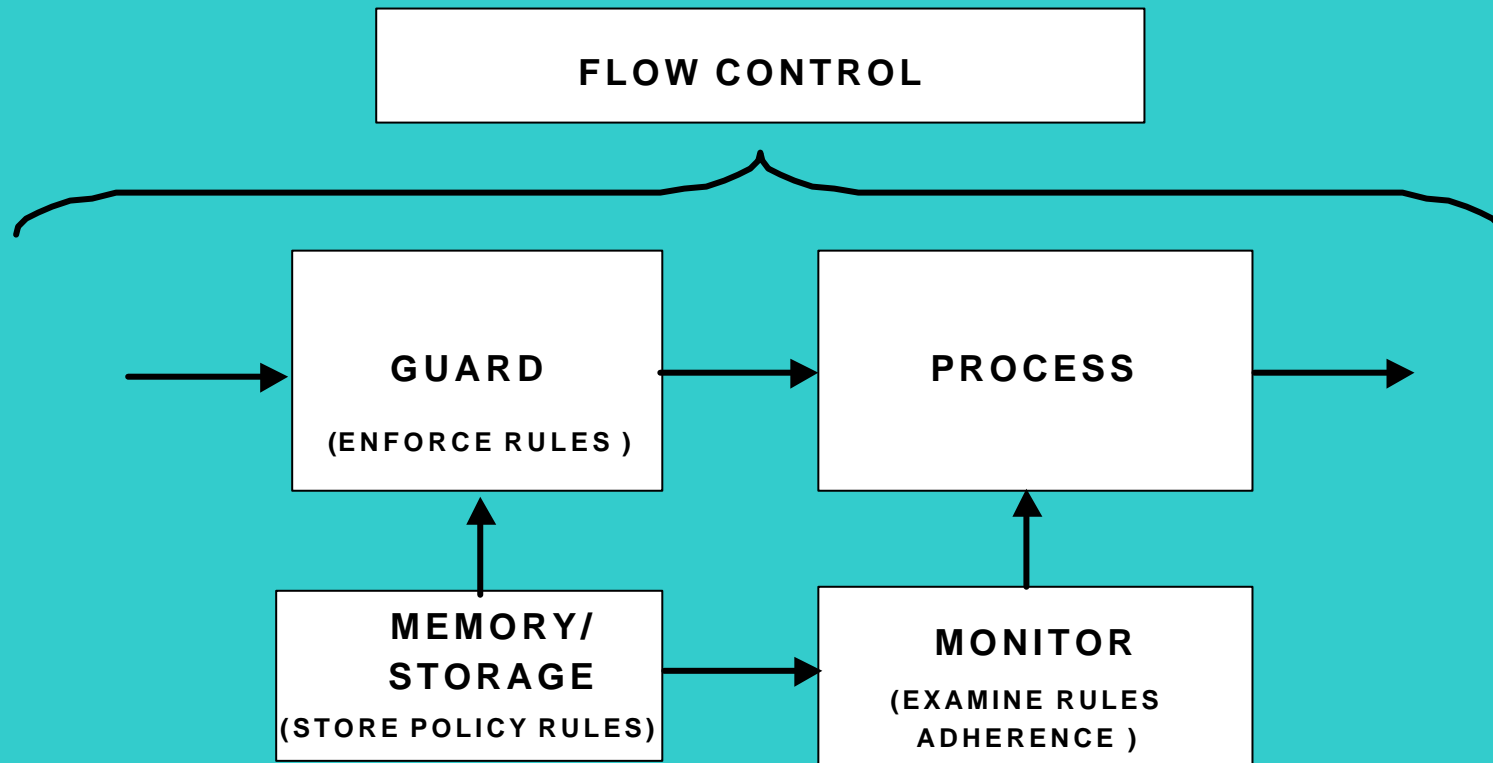
Implied

- Data Separation
- Flow Control
- Protected Storage

SCA Security Supplement

- SCA Security Supplement identifies 2 methods to provide computer security:
 - Monitor and guards
 - 3-tiered separation kernel
- Two methods are compatible
 - Monitors and guards are a subset of functions in the 3-tiered separation kernel

Guards and Monitors



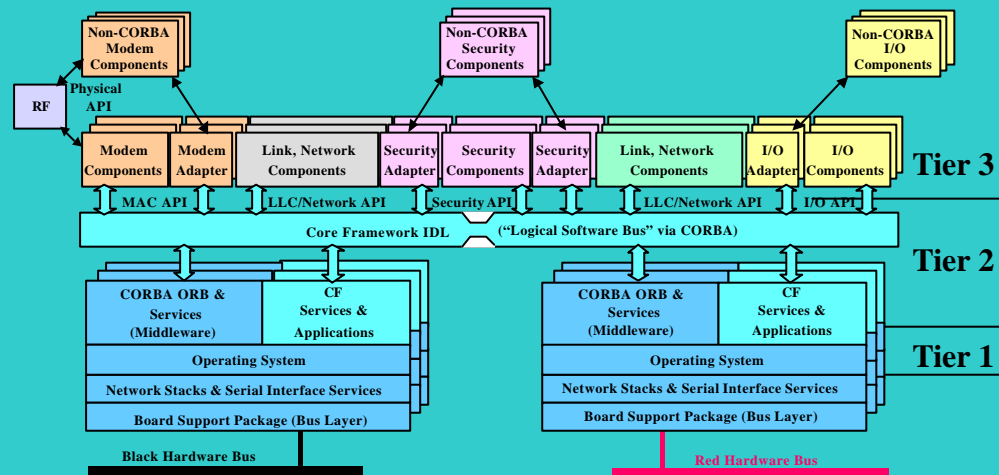
Guards and Monitors

- Guard is an in-band function through which target data passes for examination of some form
- Examples:
 - HCI security enforcement mechanism
 - Bypass Mechanism
- Monitor examines results of another process to see if expected outputs are achieved
- Example:
 - Intrusion detection system
 - Watchdog timer

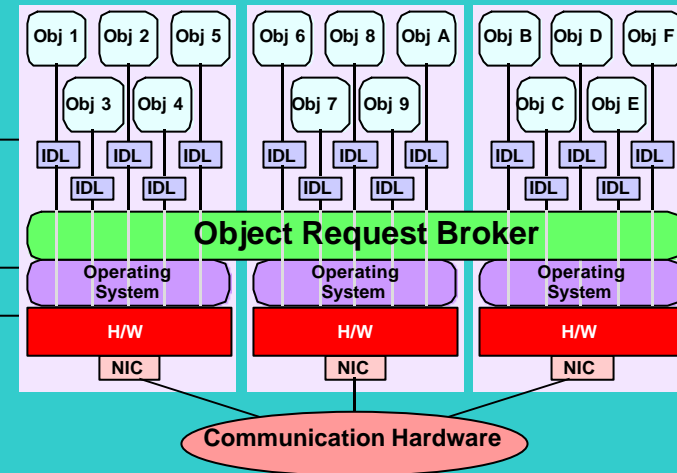
Security Architecture Outline

- Transform SCA defined Operating Environment (OE) into a Trusted Computer Base
- Defense in depth concepts applied in embedded environment
- 3-tiered separation kernel
 - Tier 1: Operating System (kernel)
 - Tier 2: Middleware
 - Tier 3: Application (e.g. Core Framework and waveforms)
- Higher tiers dependent on the correct operation of lower tiers

Software Structure Comparison



JTR SW Structure



RT COE SW Structure

Middleware: Security Enforcement

- Isolate application launch mechanisms to protected processes
- Protect object references
- CF application launch provides monitor function
- Current ORB technology security issues:
 - Places the ORB within the application process space
 - Places security services with ORB
 - Subjects security services to tamper/bypass by user applications

Benefits of Separation Kernel Security Architecture

- Simplifies transition from system high to MLS
- Partitions security enforcement support to layers
- Allows layers to be evaluated separately
- Higher assurance layers may be incorporated without impact to certification of other layers
- Supports new application security mechanisms without impacting the assurance of other enforcement mechanisms within the system

Conclusions

- Each layer enforces its own portion of the security policy
 - OS: process separation, inter-process information flow
 - CORBA: object separation, inter-object information flow
 - Application: application-specific policy enforcement
- Application becomes a full partner in defining and enforcing application-specific, security policies
- Layered security enforcement simplifies security evaluations

Cautions

- You've carefully thought out all the angles.
- You've done it a thousand times.
- It comes naturally to you.
- You know what you're doing, its what you've been trained to do your whole life.
- Nothing could possibly go wrong, right ?

Think Again.



Future Directions

- Work with business, industry and government
- Develop alternate methods of obtaining security requirements
- Develop RT security API requirements
- Involve and coordinate with RT/OS vendors to provide solutions
- Provide security environment specifications