



# Supporting Secure, Distributed, Real-time Systems: A Middleware Perspective

---

## Objective Interface

13873 Park Center Road, Suite 360

Herndon, VA 20171-3247

**Bill Beckwith**

703-295-6519 voice

703-295-6501 fax

[http://www.ois.com/  
bill.beckwith@ois.com](http://www.ois.com/bill.beckwith@ois.com)



# Topics

- ◆ Viewpoint
- ◆ Goals
- ◆ Terms
- ◆ Motivations
- ◆ Embedded Security Concerns
- ◆ Trust Model
- ◆ Proposed Solution Space
- ◆ Venue



# Viewpoint: Who are you?

- ◆ **Objective Interface Systems, Inc.**
  - Produce real-time communications frameworks
  - Lead real-time and embedded ORB vendor
  - ORB*express* product popular in
    - Defense
    - Telecommunications
    - Consumer electronics
    - Digital entertainment



# Viewpoint: Why are you doing this?

- ◆ **Address customer need for a security architecture that is:**
  - Standards-based
  - High-performance
  - Lightweight
  - Economical
  - MAC-capable
  - Cross-platform
  - Distributed
  - With minimal trusting model



# Goals of this Presentation

- ◆ Present motivations for developing a new security specification
- ◆ Solicit venues for specification effort
- ◆ Inspire discussion on potential solutions

## ◆ Authentication

- Verifying that the accessor of data or sender of a message is authentic

## ◆ Access Validation

- Providing application developers with APIs to validate that an accessor can access information

## ◆ Confidentiality

- Third parties cannot access information
- Encryption commonly used

## ◆ Encryption

- Scrambling data
- Allows transmission of information through untrusted areas

## ◆ Integrity

- Assurance that third parties have not modified information

## ◆ MAC - Mandatory Access Control

- Access controls that prevent a user from making information available arbitrarily
- As opposed to discretionary access controls like the ACL mechanism
- Typically implemented with information *labeling*

## ◆ TCB - Trusted Computing Base

- “The totality of protection mechanisms within a computer system -- including hardware, firmware, and software -- the combination of which is responsible for enforcing a security policy.” – **TCS**  
**Eval Criteria**

## ◆ Existing O/S security architectures

- + Don't require trusting the application code
- Inconsistent support across O/Ses
- Don't distribute
  - ❖ Creates huge hole in TCB when IP is used
- Most don't support MAC (Mandatory Access Control)
- Aren't extensible

## ◆ Existing distribution security architectures

### ○ CORBA Security

- + Independent standard
- + Distributes well (delegation)
- Requires trust of ORB, application, and transport (i.e. ORB, application, and transport are part of TCB)
- No MAC

- ◆ Existing distribution security architectures (cont.)
  - Banking/ATM
    - Visa Security Module
      - ❖ Cryptoprocessor
      - ❖ Protect PINs transmitted over ATM networks
        - Requires specialized, proprietary hardware
        - High performance was not a design goal
  - ???
    - Trusted RDBMS?

- ◆ **Budding applications & security architectures**
  - Information enabled warfighter
    - DoD: Joint Tactical Radio System, connected intel, ...
    - Air Force: JSF, F-22, UAVs, ...
    - Army: Objective Force, Land Information Warfare Activity, ...
    - Navy: Aegis, DD-21, ...
  - Digital entertainment content protection
    - 4C
  - Home automation security
    - Honeywell GHS



# Embedded Security Issues

## ◆ Performance Concerns

- Slower kernel operations
- Potential source of jitter

## ◆ Footprint Concerns

- Larger kernels
  - Authentication
  - Authorization
  - Encryption
  - Auditing
- Larger applications
  - Access checks
  - Error handling
  - Maintaining levels

## ◆ Application Trusting Model

- Applications part of TCB
- Scope of trust includes:
  - Application programmer
  - Tools vendors
  - O/S vendors
  - Hardware vendors

## ◆ Kernel Trusting Model

- Applications must obey security system
- Existing kernel based trust models are ignorant of distribution

## ◆ Hardware Trusting Model

- Kernel and applications must obey security system

## ◆ General standards

### ○ CORBA Security

- + Independent standard
- + Multiple implementations exist
- + Robust model for building trusted applications
- Requires trust of ORB, application, and utility libraries
- Trust is transitive
- Large implementations
- Existing implementations introduce large overhead to ORB

### ○ SSL, SSH, RSA, Triple-DES

- + Content protection
- Large implementations
- Heavy computational requirements
- Incomplete security model
- No provision for Mandatory Access Control (MAC)



# Proposed Solution Space

- ◆ Extensible security architecture
- ◆ Security system is pluggable
- ◆ Kernel security hooks
- ◆ Application access validation API
- ◆ Labeled messaging protocol and API



# Kernel security hooks

- ◆ **Allow third-party installation of**
  - Access validation
  - Auditing
  - Authentication
  - Authorization
  - Encryption
  - Information Labeling



# Application access validation API

- ◆ Used by application developers
- ◆ Consistent API between
  - Security system and
  - Application



# Labeled, messaging protocol and API

- ◆ **Below-the-middleware technology**
- ◆ **Integral with kernel security plug-ins**
- ◆ **Potential uses:**
  - Intersystem communications
    - Not limited to just IP
    - Provides for assurance, authentication, identification, confidentiality, integrity, etc. of messages
  - Peripheral communications
    - Disk storage of secure data



# Venue for specification

- ◆ Open Group?
- ◆ OMG?
- ◆ Other?



# Contact details

**Bill Beckwith**

**Objective Interface Systems, Inc.**

**[bill.beckwith@ois.com](mailto:bill.beckwith@ois.com)**

**703-295-6519**

**Real-time CORBA Forum**

**<http://www.realtime-corba.com>**