PRISMTECH
Productivity Tools & Middleware

Modeling and Implementation of Inter-Component Security for Effective Security Policy Enforcement and Channel/Process Separation

OMG's Third SBC Workshop: *Realizing the Vision*, March 6, 2007

Outline of the Presentation

- Software Defined Radio security aspects from a middleware and modeling/development tools perspective
- Security-defined deployment of components
- The need for inter-component security
- Inter-component security enforcement
- Security design support in the modeling phase
- Discussion

Software Defined Radio security

- from a middleware and modeling/development tools perspective -

- Focus of our ongoing work on software defined radio (SDR) security is on:
  - Software Communications Architecture (SCA) compliant radios
    (next stage: OMG Software Based Communications specs.)
  - Security architecture that enables high levels of assurance
- An SDR is an automated information system (AIS) that must ensure for the processed communicator information:
  - **Confidentiality**
  - **Integrity**
  - **Availability**
- The integrity of the operational state of the SCA component instances is a prerequisite for the security (esp. availability) of the communicator information
- The information processed has potentially different classifications (separation of communication channels is essential)

Channel separation

- SDR security aspects from a middleware and modeling/development tools perspective -

- Most SDRs are multi-channel radios
  - with the channels processing information belonging to different transactions and potentially classified different security levels
- Typical approach to isolate the processing of information belonging to different channels from each other is separation
  - processing on separate processors (processor lines / boards) per channel, or
  - processing in separate partitions on a single processor with a MILS OS
- Typical approach to isolate the communication (between SCA components deployed on different processors) of information belonging to different channels from each other is separation of the connections between processors (or partitions)
  - physical (separate buses per channel), or
  - common (RED) bus with Guards
  - MILS Partitioning Communications System (PCS)

Security-defined Component Deployment

- SDR security aspects from a middleware and modeling/development tools perspective -

- The SCA provides full support for statically defined allocation of SCA application components to devices (logical devices representing physical devices)
- The SCA also provides support for dynamically defined allocation of SCA application components dependent on the available capabilities of the target platform
- The allocation of SCA application components is, however, also dependent on the security classification of the channel/transaction they are intended to support.
  - Main issue is the co-location of components belonging to the same (or different) channels/transactions and/or security levels (of the transactions)
- Statically defined allocation of SCA application components to devices can take into account security constraints (requiring knowledge of the platform and the use case)
- Dynamic allocation also possible if Domain Profile can be extended.

The Need for Inter-component Security

- All interfaces (base application interfaces and *provides* ports) of an SCA component (resources, framework control interfaces, framework services interfaces) are effectively CORBA interfaces. Every CORBA interface is effectively accessible for any entity with access to the bus on which the CORBA interface is accessible.
- Control components (Domain Manager, Application Factory) typically have access to SCA components belonging to different channels (and even across the Cryptographic Boundary).
- Problem:
  Typically, a CORBA interface of an SCA component is accessible via a bus to which more entities than just the legitimate user entities have access. If the access to the CORBA interfaces provided is not controlled, all entities that have access to the bus must be trusted that they do not illegally invoke operations on CORBA interfaces (high evaluation effort).

  Otherwise, the integrity of the channel processing for which the SCA components with the unprotected CORBA interfaces are needed cannot be guaranteed. The integrity of the channel processing, however, is a prerequisite for the availability of the communicator information to its users (possibly integrity and confidentiality too).

Approaches to Inter-component Access Control

- Inter-component security enforcement -

- Availability of CORBA object references only to authorized entities (e.g., Domain Manager, Application Factory), implemented through access control by the Naming service
  - Assumes that object references cannot be learned or guessed by non-authorized entities
- Server-side, caller-identity based access control at the application level (implemented as application logic)
  - Contradictory to the component approach of the SCA
- Server-side, CORBA-level access control based on formalized, declarative server-component security policy
  - Enforcement part of the middleware (CORBA) layer of the OE
  - Policy to be enforced reflects the Core Framework logic

CORBA-level, Server-side Access Control

- Inter-component security enforcement

- Policy engine makes a decision per request based on a set of information provided with the request and context information prepared in advance (typically on the start of the execution of the component)
  - Target component identity
  - Interface
  - Operation
  - User component identity

  against a component security policy that basically is an access control list per CORBA interface
- (optionally) additional conditions that compare current local context against expected values

Enforcement of CORBA-level, Access Control

- Inter-component security enforcement

- CORBA request message interceptor as enforcement point
  - Traditional CORBA message interceptor approach
  - security enforcement point is in the same process space as application/component and possible other components
  - “lightweight“: low latency, no additional communication point or processor
- GIOP security gateways as enforcement point
  - Gateway can be deployed in a separate process space
  - Can guarantee NEATness of security enforcement at the middleware level (non-bypassable, evaluatable, always invoked, tamperproof)
  - Costs: adds latency, potentially requires separate processor/partition
- Both enforcement types interpret and enforce the same type of formal security policy per component interface
- Choice of enforcement type is determined by assurance levels of all software in the process space of the protected component

CORBA-level Security Enforcement

- Inter-component security enforcement

- Additional security function provided by the security enforcement points:
- Security audit (non-repudiation)
  - Level of detail configurable
  - Detailed information for analysis/forensics
- Access control using state (e.g., recognition of repeated illegal access attempts by corrupted components)
- Gateways can be enhanced with covert channel analysis for interactions of CF control components with SCA components belonging to different channels

Security Design Support in the Modeling Phase

- The (generic) enforcement points are part of the platform (OE).
- The platform and application specific security configuration (allocation of SCA components, formalized declarative component security policies, security enforcement mechanism) reflects the security view of the model of a waveform (or a number of waveforms) deployment on a platform.
- The “security configuration“ parts of the Domain Profile can be automatically generated (security enhanced domain specific modeling).
- The “security configuration“ parts of the Domain Profile together with the enforcement points can be evaluated if the resulting installation is compliant with higher level security policies.

PRISMTECH
Productivity Tools & Middleware

Modeling and Implementation of Inter-Component Security for Effective Security Policy Enforcement and Channel/Process Separation

OMG's Third SBC Workshop: *Realizing the Vision*, March 6, 2007