Security Modeling with AADL in the Generic Modeling Environment

Matt Eby
Institute for Software Integrated Systems
Vanderbilt University
Overview

- The Generic Modeling Environment (GME)
- AADL in GME
- Role Based Access Control between AADL Components
- Multi-Threaded C Code Generation
- System Demo
- Security Testbed
Background and Motivation

• Team for Research in Ubiquitous Secure Technology (TRUST)
  - NSF cybersecurity Science and Technology Center
• Security in Networked Embedded Systems
• Integrating Security Modeling and Analysis Tools into existing Toolchains
• How are security policies on the model level enforced?
Generic Modeling Environment (GME)

- GME is a configurable toolkit for creating domain-specific modeling and program synthesis environments.
- MetaGME = EMF-Ecore
- Graphical Editor comes with MetaModels specified in GME
- GME has Builder Object Network (BON)
  - Code Generation
  - Analysis
Current Status of AADL in GME

• AADL concepts currently implemented
  - **Software Components**
    - Data, Subprograms, Threads, Thread Groups, Processes
  - **Execution Platform Components**
    - Processors, Memory, Buses, Devices
  - **Features and Shared Access**
    - Ports, Port Groups, Subprograms
  - **Connections and Flows**
    - Port Connections, Parameter Connections, Access Connections, Flow Specifications and Implementations, End-To-End Flows

• **Future Work**
  - Modes, Properties
Integrated Co-design Environment

- Domain-specific Modeling Languages (AADL, Simulink/StateFlow, ...)
- Security modeling for different platforms
- Model Analysis tools
- Code Generators
Current Tool Status

• Textual AADL Generator
  - Weather Predictor Example (Pages 58-60 in AADL Standard Document)
  - Future Work:
    • Upgrade to Current MetaModel

• C Code Generator
  - Multi-Threaded C code with Makefile
  - Examples:
    • DC Motor Controller
    • Three Tank System
  - Future Work:
    • Large Scale Distributed Control
Security Mechanisms - RBAC

- Component level access control - Role Based Access Control
  - Authentication between Software Components
  - Components are assigned roles and permission requirements
  - Permissions are mapped to roles
Enforcing Security Mechanisms

- Partitioning - embedding components in separate domains
  - **Linux Security Mechanisms**
    - Gentoo 2005.1 with 2.4.32 kernel
    - Grsecurity enhanced kernel
    - Kernel working in privileged mode (ring0)
    - Userspace programs working in unprivileged mode (ring3)
    - User identifiers UID
    - Group identifiers GID
    - chroot()
Security Mechanisms

```adl
data VoltageLevel
end VoltageLevel;

subprogram DACSetVoltage
features
  voltage: in parameter VoltageLevel;
  chan: in parameter channel;

end DACSetVoltage;
bus VME
end VME;
device device
features
  BusConnection: requires bus access VME;
  device;
memory Memory
features
  BusConnection: requires bus access VME;
  Memory;

data Channel
end Channel;
subprogram DACsetup
dend DACsetup;
processor Processor
features
  BusConnection: requires bus access VME;
  Processor;
thread implementation GetDataThread
calls { getdata: subprogram getdata;
}
end GetDataThread;
thread implementation ListenThread
calls { netlisten: subprogram netlisten;
}
end ListenThread;
process implementation Gateway
subcomponents
  thread GetDataThread;
  thread ListenThread;
end Gateway;
processor implementation Processor
end Processor;
thread implementation CommThread
calls { startserver: subprogram startserver;
}
end CommThread;
thread implementation MainThread
calls { DACsetup: subprogram DACsetup;
  runloop: subprogram runloop;
```
Demo - System Level
Demo – Process Level
Demo – Permission Requirements
Demo – Role Assignment
Demo – Permission to Role Mapping
Demo – Subprogram Level
Testbed Configuration

Different SW platforms:
- Linux + GRSecurity
- Others (LynxOS, VxWorks,..)

PCI-DDA08/12
Data acquisition board

xPC

Wireless Link

DAQ

Controller

Controller

Controller

Plant Simulator

Single board computer
SBC4495 from Micro/Sys
1. Three tank control system model

2. Code generation

3. Deployment environment

4. Network attack on controller
Future work

• Modeling different security aspects: access control, security measures, confidentiality, data leakage, privacy, attack trees
• Creating toolchains for complex security analysis and system deployment
• Application for security testing
Questions?

- Security is a growing concern in embedded systems
- Security needs to be provisioned in system design models
- Model-driven approach has potential in addressing security concerns