MILS Research Montage

LAW
Work-In-Progress Session
December 6, 2011

Rance DeLong
Consulting Researcher
MILS Efforts Overview

- MILS Vision
  - Constitution
  - Concepts
  - Manifesto
  - Example
  - Dissemination
    - NSA
    - TOG
  - MIP
    - MIP
    - Guard
    - SKPP
    - MNSPP
    - MCSPP
    - AADL
  - Effort Categories
    - Efforts/Results to date
    - Future
    - Sponsored by AFRL / CMPO

- Effort Categories
  - *Sponsored by AFRL / CMPO

- SIs / Programs
  - RTI
  - Galois
  - RCI
  - OIS
  - GHS
  - LW
  - WRS

- Standards
  - API
  - DCI

- Implementation
  - Patterns
  - Assemblies
  - Tools
  - Ref Impls

- Math
  - Lecture Notes
  - Found’nm Comps
  - Science
  - Science

- Eval & Cert
  - Scheme
  - Evidence
  - Assur. Case

- Compos. Cert.
  - Compos. Cert.

- Found’nm Comps
  - Opera’nl Comps

- Dissemination
  - LAW
  - TOG
  - ICCC

- Example
Research Enabling MILS Development and Deployment (REMDaD)*

- **Objective:**
  Move to next stage of MILS deployment and development

- **4 Themes**
  - Components – development and assurance of individual components
  - Integration – integration of MILS components and systems
  - Deployment – facilitate MILS deployment
  - Certification – enable MILS evaluation and certification

- **Initial tasks (2010)**
  - Evidence and toolchains for MILS certification study
  - MILS Cross Domain Solution (CDS) operational component Study
  - MILS Delivery, Configuration, and Initialization (DCI) Study

* Performed at SRI, sponsored by AF Research Laboratory and AF Cryptographic Modernization Program Office.
R. DeLong
Research Enabling MILS Development and Deployment (REMDaD)*

- Current tasks (2011-2012) -
  (John Rushby, Dave Hanz, Rance DeLong)
  - AADL and MILS
  - MIPP completion (MIPP as a document)
  - “Programming the MIPP” (MIPP encoded in the CCAE)
  - MILS Delivery, Configuration, Initialization model
  - MILS Cross Domain Solution investigation
  - MILS Network Subsystem Protection Profile

* Performed at SRI, sponsored by AF Research Laboratory and AF Cryptographic Modernization Program Office.
MILS is based on composition of cooperating components defined by related Protection Profiles*

- Separation Kernel (SKPP)
- MILS Network System (MNSPP)
- MILS Console System (MCSPP)
- MILS Extended Attributes PP (MEAPP)
- MILS File System (MFSPP)
- . . .
- MILS Integration Protection Profile (MIPP)

[Diagram showing relationships between SKPP, MNSPP, MCSPP, MEAPP, MFSPP, and MIPP]
Mils PPs are expected to achieve this:

System A
- SK_4
- MEA_2
- Console_1
- Network_3
- File System_3

System B
- SK_1
- MEA_3
- Console_4
- Network_1
- File System_4

! = It works!
Illustrative Architecture of a MILS-based MLS workstation - a collection of connected “things”
Architecture of a MILS based workstation - itself is *Something*

*Something that must be designed.*
*Something that has properties.*

Architecture as an *Integration Framework*
This **Something** is what the MIPP describes

- The system level **security problem** (T/P/A)
- The system level **security objectives**
- The system level **SFRs** and **SARs**
- A system concept and **reference architecture**
- Identification of, and connections among, the **components**
- A basis for formal **composition** of component properties
- **Constraints** on the MILS components that fit in the “holes”
  - Security objectives, or modified ones, that pass to the component
  - Relationships and obligations (rely-guarantee) among the components
  - Interaction schemas for interacting components
Some architecture alternatives for MILS network system

**Nothing Trusted**

**Everything Trusted**

- Code manipulates data in multiple security domains
- Individual data items associated with a single security domain

**Combination of Trusted and Untrusted**

- Individual data items associated with a single security domain

R. DeLong
System Inputs, Outputs, Relies and Guarantees

HIGH inputs \[ \rightarrow \] Relies \[ \rightarrow \] HIGH outputs

LOW inputs \[ \leftarrow \] \[ \rightarrow \] LOW outputs

Guarantees \[ \leftarrow \]
MILS System from Components/Subsystems

Constraints:

H(HI,H0) → Relies → Guarantees → OOOOOO...

HIGH Inputs

L(LI,LO) → LOW Inputs

S(HI,H0,LI,LO) → LOW Outputs

HIGH Outputs

Properties: P(HI,H0,LI,LO) st S ≤ P
Compositional Relies / Guarantees

a) A → C

b) A → C

c) Relies → Guarantees

S
MILS Composite Assurance Case

- Compose assurance cases using Assume-Guarantee Reasoning
- Assumptions from MI assurance case become requirements on the components
- Assured Claims from component assurance cases become evidence for MI
Common Criteria Authoring Environment as a distributed collaboration environment
CCAЕ User and Components

Rule Base
- CC Component Operation Rules
- Semantic Rules
- Relational Model
- Workflow Rules

Environ Library
- Components, CC SFRs/SARs, Interps, CIM, Security Ontology, Resource Registry, MILS Integ FW

Author/Reviewer
- Parent PP, MILS TOE Concept, or TOE Flow-down Requirements

Document Creation/Revision
- Doc Assembly, Catalog Selection, Checking, Reviewing, Inference, Rule Execution, Queries, XML gen

XML

CCAЕ Document Repository

Document Factbase

UI Agent

Doc Creation Library
- Conventions, Doc comp classes, Doc generators: PP, ST, FSP

Document Publishing
- Rendering & Conversion

Documents & Reports
- PDF, DOCX, XLSX, ...

Project Team Exchange or Export

PP, ST, stats
Relational Structure of a Protection Profile

```
\text{Assumptions} \quad \text{A}

\text{Policies} \quad \Pi

\text{Threats} \quad \mathcal{T}

\text{Security Objectives} \quad \Omega

\text{Functional Requirements} \quad \text{SFR}
\begin{align*}
\text{FAU, FCO, FCS, FDP,} \\
\text{FIA, FMT, FPR, FPT,} \\
\text{FRU, FTA, FTP}
\end{align*}

\text{Assurance Requirements} \quad \text{SAR}
\begin{align*}
\text{ACM, ADO, ADV,} \\
\text{AGD, ALC,} \\
(\text{AMA}), \text{ATE, AVA}
\end{align*}

\text{Environment Requirements}

\text{"Space" of PPs} = (2^\mathcal{T} \times 2^\Pi \times 2^\text{A} \times \Omega \times 2^\text{SFR} \times 2^\text{SAR})
```
Approximation of a MILS PP Oracle ($M_{C\text{CAE}}$)

$PP = (2^T \times 2^\Pi \times 2^A \times \Omega \times 2^{SFR} \times 2^{SAR})$

$M_C$ a candidate member of $M$

CCAE drives $M_C$ toward $M$ by measuring consistency and coverage with respect to $M_{C\text{CAE}}$

$E \subseteq PP$ evaluable PPs

$M \subseteq E$ MILS evaluable PPs
Projecting the MILS PPP to standard PPs

Projection Function

Polymorphic PP with sub-profiles A, B, C

Projection Function

\[ f \, \text{PPP}_{ABC} \{ \{A\}, \{A,B\}, \{A,C\}, \{A,B,C\} \} \]

= \{ PP_A, PP_{AB}, PP_{AC}, PP_{ABC} \} 
+ Evaluation Work Unit Checklists

Difference operator “\” applies comp’nt dependency, hierarchy, and other PP property closures. Differential work units assume ordered evaluation of PPs.
Entailed work units to be performed to evaluate $\int PPP_{ABC} \{A\} = PP_A$

Note, the following Venn diagrams represent contents of projected PPs, not PPP sub-profiles. Projected PPs may have substantial intersection, while sub-profiles may be disjoint.
Evaluation differential work units (2)

Work units entailed to evaluate $\int PPP_{ABC} \{A,B\} = PP_{AB}$

$PP_{AB}$ common work units completed for evaluation of $PP_A$

Differential work units $AB \setminus \{A\}$ to be performed to complete evaluation of $PP_{AB}$

Work units already completed during evaluation of $PP_A$
Evaluation differential work units (2)

$PP_{ABC}$ common work units completed for evaluation of $PP_A$ and $PP_{AB}$

Differential work units $ABC \setminus \{A,AB\}$ to be performed to complete evaluation of $PP_{ABC}$

Work units entailed to evaluate $\int PPP_{ABC} \{A,B,C\} = PP_{ABC}$
Generalized Delivery, Configuration, and Initialization interpretation

- Interleaved configuration and delivery
- Configuration and integration is **incremental** due to separation of concerns and separation of duty
- OEM TOE developer is responsible for providing trusted delivery and for trusted initialization
- Trusted delivery should protect TOE to the deployment environment, providing basis for establishment of secure initial state
- There can be multiple intermediate integrator environments!
Incremental accumulation of component / configuration data bundle protected by, and updated within, Trusted DCI pipeline

Components

- fs
- pcs
- net
- con
- ap1
- ap2

Applications

- deploy_env
- init
- cd
- sk
- con
- pcs
- net
- ap1
- ap2

Configuration actions

- bundle
- sk
- c1
- c2
- c3
- c4
- c5
- c6
- c7
- cn
- cm
The big picture, scope of phases
Temporal overlap and location spanning

Development Env

Delivery

Integration Env(s)

Initialization

Reconfig

Configuration

User Env

Operation

Developer Environment Integrator Environment(s) User (deployment) Environment
Generalized Reconfiguration

System Configuration Property $\Phi$

Interval Configuration Properties

Trace of System States

Operational Interval 1

Operational Interval 2

$\Phi$ - system configuration property

$\phi^i$ - interval configuration property

$\tau_R$ - reconfiguration transition

R. DeLong 26