EURO-MILS: Building and certifying modular secure systems

Sergey Tverdyshev, SYSGO
The Euro-MILS consortium

MILS Workshop 2015
20.01.2015 Amsterdam

www.euromils.eu
EURO-MILS Consortium

14 Partners from 6 Countries

- TECHNIKON
- SYSGO
- AIRBUS GROUP
- DFK
- UNIVERSITEIT GENT
- OPEN SYNERGY
- THALES
- UYENSITÉ PARIS SUD
- JEMM research
- T-SYSTEMS
- Open Universiteit
- Innovations, Germany
- Innovations, France
- Innovations, Germany
- Innovations, France
EURO-MILS: Strategy and Objectives

- High-criticality networked cyber-physical systems
  - Drivers are avionics and automotive
  - EURO-MILS delivers cross-domain solutions

- Integration and networking requires trustworthy ICT

- MILS Architecture
  - High-assurance security architecture
  - Scalable and affordable security
  - Compositional design, assurance, security

- EURO-MILS: European MILS architecture and certifiable platform

ICT: Information and Communication Technologies
MILS: Multiple Independent Levels of Security
COMPOSITIONAL SYSTEM DESIGN
FOR SECURITY AND SAFETY
Developing System Architecture

• System is
  • a group of related components that work together
  • possessing a set of properties

• To bring that components to life you need an execution platform
  • Execution platform introduces new components and interfaces
  • Execution platform has (physical) resources
  • Execution platform possesses a set of new properties
  • i.e. refine system design
Developing System Architecture

- **Generic problems:**
  - Composition preserving safety, security, assurance arguments
  - Refinement is a composition
  - Mitigate effects of “have to refine”
    - where we need something to execute systems
MILS Architectural Approach

Application plane

Refinement

MILS induced abstraction

Resource plane

MILS Platform (Separation Kernel)

Hardware (CPUs, memory, and devices)
MILS induced abstraction enables truly **compositional**

- Safety and Security
- Assurance
- Evaluation
MILS DESIGN AND ASSURANCE FRAMEWORK
MILS Design and Assurance Framework

• **EURO-MILS** focus is to create a framework with focus on
  • Compositional Design/System integration
  • Compositional Assurance
  • Certified MILS separation kernel

• Framework shall cover major life-cycles of system design, integration, validation, evaluation

• **EURO-MILS** validates framework on industrial applications in avionics and automotive

• **Goal:** create validated MILS Framework as set of
  • specifications, examples, guidelines,
  • evaluation methodology
  • to ease system designing and creating assurance artefacts
MILS is not equal to separation kernel (SK)
- MILS SK cannot be a stand-alone component neither in application nor in certification (PP)

MILS is
- Design approach and Architecture
- System integration approach
- Mils API
  - see also The Open Group MILS WG
- High-assurance components (separation kernel, minimal file system, network etc.)
  - ...

However, one of the cornerstone is a separation kernel
MILS Framework

MILS Architecture Template

- Development
- Assurance

Formal methods for components and system integration
Formal methods in Common Criteria

MILS Vulnerability analysis
High-assurance methods

Protection Profile for Separation Kernel
Formal Interfaces and Components specs,

MILS Compositional Evaluation Methodology
T-puzzle Compositions Specifications

Domain: Avionics
Domain: Automotive

Assurance

Common Criteria MILS Compositional Evaluation Methodology
MILS Framework

Development

MILS Components
- Functional Specification
- Separation Kernel
- Partitioned File System
- Security Audit
- Policy enforcing HW/drivers

MILS/MLS Systems Integration
- System Integration Guidelines
- Domain: Avionics
- Domain: Automotive

Assurance

Common Criteria
- T-composition
- Puzzle-Compositions

MILS Compositional Evaluation Methodology
- Protection Profile for Separation Kernel
- Formal Interfaces and Components specs...

Specifications
- Formal methods for components and system integration
- Formal methods in Common Criteria

MILS Vulnerability analysis
- High-assurance methods
MILS Framework: Developer track

- MILS architectural template
- Integration Guidelines
- Separation Kernel
- System MILS architecture
- System Components
- Implementation
MILS Framework: Assurance track

- State of the art
  - PP4SK, OSPP, SKPP, TEE, FM...
- Integration Guidelines
- High-Assurance Qualification Methodology
  - AVA_VAN.5 Methodology
  - Formal methods
- PP for Separation Kernel
- Domains
  - Good-practices
- System MILS Architecture and Implementation
- Assurance
The Developer Track
MILS architectural template defines main components. Example: Separation Kernel (SK).

- *Separation in space* of applications hosted in different partitions from each other and from the separation kernel

- *Separation in time* of applications hosted in different partitions from each other and from the separation kernel

- *Provision and management of communication objects*

- *Management* of and access to the SK and SK data

- Separation kernel *self-protection and accuracy of security functionality*

- *Generation and treatment of audit data according to the configuration*
Avionics

Automotive

Trustworthy ICT for networked high-criticality systems
Example: Aircraft Security Domains

Picture adapted from ARINC 811.
Domains are defined in ARINC 664 Part 5.
**The Avionics MILS Gateway**

**Key aspect of architecture:**
Rely on MILS platform security services for the implementation of gateway layers (e.g. coarse information flow control of separation kernel and using unidirectional flow).

Components required for processing the data stream between the two domains:

- Receiver Component
- HTTP Chain
- Transmitter Component
- NIC
- Transmitter Component
- TFTP Chain
- HTTP Chain
- NIC
- Receiver Component
- HTTP Chain
- TFTP Chain
- Common Components
  - Stateful Context Manager
  - Auditing & Alerting
- Monitor

Interfaces for communication outside the domain
Bidirectional communication link
Target of automotive security measures is the protection of instrument cluster and head unit display control, as well as the underlying virtualisation platform. Under no circumstances, these units may be compromised or disturbed in their normal operation.
Automotive Telematics Environment

**Backend server**
- Target connection
- Online services
- Update repository

**Smartphone**
- App
- App

**Automotive demonstrator**
- Android
- Autosar
  - Network manager
  - PikeOS
  - Modem
  - CAN

**Internet**
MILS is applicable and gathering interest across all domains

- Avionics/UAV
- Automotive
- Industrial automation
- Railway
- Railway automation
- Mobile devices
- Telecom and communication
- Multiple-payload satellites
- Sea/Subsea
- Banking
- ...

More Use-cases
The Assurance Track
EURO-MILS Platform: Common Criteria Certification

An international standard (ISO/IEC 15408) for computer security certification

EAL: Evaluation Assurance Level

- **EAL 1**: Functionally Tested
- **EAL 2**: Structurally Tested
- **EAL 3**: Methodically Tested and Checked
- **EAL 4**: Method Designed, Tested and Reviewed
- **EAL 5**: Semiformally Designed and Tested
- **EAL 6**: Semiformally Verified Design and Tested
- **EAL 7**: Formally Verified Design and Tested

**EURO-MILS Project Goals EAL 5+ (7)**

Certification Schemes
- ANSSI (FR) and BSI (GE)

**Protection Profile**
- Security objectives
- Security requirements
- Equivalent or more restrictive
- All

**Security Target**
- Security objectives
- Security requirements

**Target of Evaluation**

TOE meets targets and meets security objectives.
MILS architecture is the enabler for high-assurance compositional certification.
The core is Separation Kernel.
Components under certified composition:
- Hardware, Separation kernel, Applications.
Compositional Certification: Scenario-P

- Puzzle Composition
  - Exchange system component with interface/function-compatible one
  - Use-cases
    - Product from Vendor-A is replaced by product from Vendor-B
    - Flexible in-the-field update
Protection Profile for Separation Kernel

- **Protection Profile** defines a MILS separation kernel

- **Protection Profile** defines
  - a special kind of operating systems for embedded systems
  - with support for real-time

- **MILS separation kernel** allows separation of applications running on the same platform from each other
  - User applications can be malicious and be developed by arbitrary developers
TOE Physical Boundaries

User partition 1
User application(s)
E.g., C application

User partition 2
User application(s)
E.g., POSIX

User partition N
User application(s)
E.g., Linux
TOE Physical Boundaries

User partition 1
User application(s)
E.g., C application

User partition 2
User application(s)
E.g., POSIX

User partition N
User application(s)
E.g., Linux

Hardware

Firmware

Bootloader
TOE Physical Boundaries

- User partition 1
  - User application(s)
  - E.g., C application

- User partition 2
  - User application(s)
  - E.g., POSIX

- User partition N
  - User application(s)
  - E.g., Linux

- User partition API

- PikeOS separation kernel

- ASP

- Hardware
  - Firmware
  - Bootloader
TOE Physical Boundaries

User partition 1
User application(s)
E.g., C application

User partition 2
User application(s)
E.g., POSIX

User partition N
User application(s)
E.g., Linux

User partition API

Commun. objects

PikeOS separation kernel

ASP

Hardware
      Firmware
      Bootloader
TOE Physical Boundaries
TOE Physical Boundaries
TOE Physical Boundaries

User partition 1
User application(s)
E.g., C application

User partition 2
User application(s)
E.g., POSIX

User partition N
User application(s)
E.g., Linux

User partition API

Audit data
Commun. objects
Conf. data
ASP

PikeOS separation kernel

PSP API

Hardware
Firmware
Bootloader
TOE Physical Boundaries
TOE Physical Boundaries

- User partition 1
  - User application(s)
  - E.g., C application

- User partition 2
  - User application(s)
  - E.g., POSIX

- User partition N
  - User application(s)
  - E.g., Linux

- System partition 1

- System partition M

- User partition API
  - Audit data
  - Conf. data
  - Commun. objects
  - ASP

- System partition API
  - System ext. API
  - System extension
  - PSP API
  - PSP

- Hardware
  - TSF
  - User partition content, arbitrary user data; communication objects content, arbitrary user data; audit data
  - TSF data, incl. amongst others, configuration data and shapes of user partitions, communication objects, system components
  - System component content, user data that has to be approved by the system integrator
  - Operational environment

- TOE boundary
System Integration and Roles

User application developers

User application 1
e.g., C program

User application 2
e.g., POSIX

User application N
e.g., Linux

System integrator

System partition content

Configuration data in textual form

TOE manufacturer

TOE binary

TOE User Manuals

Product binary image

Protection Profile AVAILABLE ON
http://euromils.eu

Parts of the TOE, provided by the TOE manufacturer

Integration tool chain, provided by the TOE manufacturer

Content of user partitions, this content can be arbitrary (from security
point of view) and also be applied by any 3rd party

Content of system components and configuration data (in textual form); these elements,
even if supplied by a 3rd party, are under sole responsibility of system integrator and shall
be approved by him/her; see OSP P.SYSTEM_INTEGRATOR below.
On-going work

- MILS Vulnerability Analysis
  - Define attack paths
    - Inspired by the SOGIS JIL SmartCard
  - Define evaluation methodology
  - Focus on system integration and composition
  - Goal:
    - Define work items for evaluators
    - Define what, at least, system integrator should consider

- MILS System Integration Guidelines
  - Good-practices on system integration
  - Examples of MILS Architecture Template applications
  - Focus on system integration and composition
  - Goal: ease the work of the system integrator
High-Assurance

FORMAL METHODS
Formal Modelling: Separation Kernel

Complex generic model - prove *once and for all* that Proof Obligations imply separation

- Formal Model of Separation Kernels (CISK)
  - Proof Obligations (a.k.a unwinding conditions)
  - Separation

Formal Model induces modelling methodology

- High Level Model of Implementations (PikeOS)
  - Proof Obligations instantiated for PikeOS

Once Proof Obligations discharged for PikeOS, Intransitive Noninterference immediately follows
Specification

- Separation property is expressed as non-interference
- Based on more than 35-years of research
  - a refinement of „industry-standard“ Rushby non-interference, extended by stateful actions
- Small, comprehensible, evaluable, trustworthy
  - This is our “gold” model, you have to have a warm feeling by looking at it 😊

- Single core model (CISK) has been published
  - AFP - Archive of formal proofs
  - AFP contains only approved theories
  - Multi-Core model is being finalized
Specification: Non-Interference

System Components

Security Policy
Formal Implementation

- **Implementation Model**
  - Model of PikeOS separation kernel actions
  - The formal implementation contains proves for the proof obligations of the specification

Formal Model induces modelling methodology

Once Proof Obligations discharged for PikeOS, Intransitive Noninterference immediately follows
definition separation :: bool where "separation ≡ ∀ u x . well_formed_executions x → 
{ x } ⊆ u { x' . well_formed_executions x' ∧ littered u x x' }"

lemma PikeOS_instantiates_CISK:
shows "Controllable_Interruptible_Separation_Kernel"
    step
    output_f
    initial_state
    current
    cswitch
    precondition
state_invariant
duration
action_sequences
aborting
waiting
involved
ifp
vpeq
dom_act_equivalent"

proof -
  write state_invariant ("\{\}" 100)
  and current ("\^\)"

  corollary instantiation_is_secure:
  shows PikeOS.separation
  using PikeOS.unwinding_implies_separation_CISK
  by blast
On-going work on a base formal model for MILS system

- Formalisation of the “MILS Architectural Template”
- Separation kernel is a component
- Express information flows on top of separation kernel
- Integrate security policies of other critical components, e.g. file system, network stack
- Target user-level security policies, e.g. re-graders with labelled information flows
High-Assurance

FORMAL METHODS AS CERTIFICATION ARTEFACT
Goal: Develop framework how to create formal models for Common Criteria evaluation

What we are doing
- Developing guidelines for developers (how to do) and evaluator (how to check) formal models in Isabelle/HOL
- Isabelle/HOL description for certification scheme
- Template to instantiate developed
  • Formal specification
  • Formal implementation
  • Formal proof
to form Common Criteria artefact (for ADV_SPM)
- Artefact compliance with AIS34 (BSI) and Note12 (France)
SECURITY VALUE?
EURO-MILS SURVEY
EURO-MILS Value?

Ecosystems
- Medical
- Defense
- Control
- Avionics
- Transport
- Smart cities
- Energy
- Finance
- Smartcards
- Smart home
- Personal
- Automotive
- Entertainment
- Telecoms
- Security

- Security and Safety –
- Certification and User acceptance –
- Virtualization and Partitioning –

Avionic Prototype
Virtualisation
Realtime OS
MILS
Automotive Prototype
Certification

Common Criteria EAL5+ Certified
EURO-MILS Social Survey

- EURO-MILS Context : Common definitions
  - Security, Safety, Trustworthiness,
  - Embedded systems, virtualization, partitioning, MILS
  - Certification, User acceptance, standards

- Christophe Toulemonde - JEMM Research
  - christophe.toulemonde@jemmresearch.com
  - +33 6 30 67 95 57

  professionals interviewed on
  - Security and Safety
  - Platform Virtualization and Partitioning
  - User Acceptance and Certification

- EURO-MILS Consumer Point of View
  - Via a Online survey of 537 respondents from 6 geographies (DE, UK, FR, IT, SP, BX)
    - Information security value, practices
    - Security and data privacy expectations and assurance
    - Acceptance of technologies and Trust
SUMMARY
EURO-MILS Main Outcomes

- Trustworthy foundations by the MILS approach, architecture, and applications

- MILS platform and its usage
  - Design, development and usage of a MILS platform based on virtualization technique
  - Framework to develop secure and safe products
  - Integrating domain-specific functionalities and components

- High Assurance
  - Certification along highest levels of “Common Criteria”
  - Pragmatic approach to use formal methods for certification
  - Innovative approach for compositional security assurance and vulnerability analysis
    - New CEM units, guidelines

- True cross European certification
  - Cross-European usage of the Common Criteria for high EALs
  - European approach for a generic certification process acceptable by national certification authorities (ANSSI, BSI)

- Validation of concepts by two prototypes
EURO-MILS CONTRACT NO: 318353

"The EURO-MILS project has received funding from the European Union's Seventh Framework Programme ([FP7/2007-2013]) under grant agreement number ICT-318353.”

If you need further information, please contact the coordinator:
Technikon Forschungs- und Planungsgesellschaft mbH
Burgplatz 3a, 9500 Villach, AUSTRIA
Tel: +43 4242 233 55     Fax: +43 4242 233 55 77
E-Mail: coordination@euromils.eu

The information in this document is provided “as is”, and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.