SysML Model Transformation for Safety and Security

Florian Lugou, Rabéa Ameer-Boulifa, Ludovic APVRILLE
ludovic.apvrille@telecom-paristech.fr

ISSA’2018 - Barcelona
Outline

Context: Security for Embedded Systems
Embedded systems

SysML-Sec
Method
SysML-Sec

Case study
Case Study

Conclusion
Conclusion, future work and references
Examples of Threats

Transport systems

- Use of exploits in Flight Management System (FMS) to control ADS-B/ACARS [Teso 2013]
- Remote control of a car through Wifi [Miller 2015] [Tecent 2017]

Medical appliances

- Infusion pump vulnerability, April 2015.
  http://www.scip.ch/en/?vuldb.75158
How to Identify Vulnerabilities?

Investigations

- Testing ports (JTAG interface, UART, . . .)
- Firmware analysis
- Memory dump
- Side-channel analysis (e.g. power consumption, electromagnetic waves)
- Fault injection
- . . .

Secure your systems!

Develop your system with security in mind from the very beginning

Our solution: SysML-Sec, supported by TTool
Firmware Dumping
Goal: Designing Safe and Secure Embedded Systems

**TTool**

- HW/SW Partitioning
- Soft. Design
- Formal Verification
- Simulation

**System specification** (includes software specification)
TTTool: Key Features

- Model-Driven Engineering tool
- Free and Open-Source
  - Plug-in can be used to insert private/commercial features
- Easy to use
- Focus on safety, security and performance
- Formal verification at the push of a button
Common issues (addressed by SysML-Sec):

- Adverse effects of security over safety/real-time/performance properties
  - Commonly: only the design of security mechanisms
- Hardware/Software partitioning
  - Commonly: no support for this in tools/approaches in MDE and security approaches
Context: Security for Embedded Systems

SysML-Sec

Analysis

Requirements

Safety

Functional

Security

Attack Trees

Fault Trees

HW/SW Partitioning

Application

Architecture

Mapping

Attacker Scenarios

Security Countermeasures

Firewall, Data Security, ...

Failsafe Mode, Plausibility Check, ...

Software Design

Security Countermeasures

Security Algorithms, ...

Verification

Safety

Performance

Security

Legend

Modeling

Verification

Security

Safety

User-defined

Automatic

Reconsideration

Fully supported by TTool
Partitioning

Before mapping

- Security mechanisms can be captured but not verified

After mapping

- Verify security (confidentiality, authenticity) according to attacker capabilities
  - Whether different HW elements are or not on the same die
  - Where are stored the cryptographic materials (keys)
  - Where are performed encrypt/decrypt operations
- Impact of security mechanisms on performance and safety
  - e.g. increased latency when inserting security mechanisms
Partitioning Verification

Modeling

Automatic Verification

Safety

Performance

Security
Security Verification

SysML Model

ProVerif Specification

Results

Translation

Verification

Backtracing

Task1

Task2

CPU1

CPU2

Bus1

Task1

Task2

sencrypt(Task1.secretData, Task1.key)
Automated Proverif Specification Generation

- **Main idea**
  - Decompose SysML-Sec behaviors into a set of *basic blocks*
  - Generate Proverif code

- **The semantic function for generating the code:**
  - Processes generation
    \[
    [[.]]^p_ε : Basic\_block \rightarrow Proverif\_process
    \]
  - Main process generation
    \[
    [[.]]_ε : SysML\_components \rightarrow Proverif
    \]
Safety and Security Mechanisms

Data Encryption/ Authentication

Safety
Security
Performance

- Data Encryption/ Authentication
- ECUCommand(1)
- chl

14/33 Sept. 2018 Institut Mines-Telecom
Safety and Security Mechanisms (Cont.)

Data Security with Hardware Security Module

- Safety
- Security
- Performance
Safety and Security Mechanisms (Cont.)

Redundancy/Coherence Check

Add security

Add security

AutonomousSystem

Perception1

Supervisor

coherenceCheck

[data1-data2 > threshold] {false)

dataValid

error

<<CPU-RR>> CPUPerc1

Perception1

<<BUS-CAN>> Bus1

<<CPU-RR>> CPUSup

<<BUS-CAN>> Bus2

Perception2

Safety

Security

Performance

16/33 Sept. 2018 Institut Mines-Telecom
Safety and Security Mechanisms

**Failsafe mode**

- `systemCheck`
  - [systemOk]
  - [else]
  - `defaultMode`
  - `failsafeMode`
  - ...
  - ...

**Safety**

- Green arrow

**Security**

- Red arrow

**Performance**

- Green and red arrows

---

Context: Security for Embedded Systems
SysML-Sec
Case study
Conclusion
Safety/Security/Performance

Requirements

Security
- Automated generation

Safety

Performance

System design

Verification of design w.r.t. requirements

Security
- Fails
  - Reconsider security req.
  - Add/modify security mechanisms
  - Modify architecture (private bus, etc.)
  - Modify mapping
  - Succeeds :-) Security leads to unsafe behaviour
  - Security leads to degraded perf. (e.g., increased mean latency)

Safety
- Fails
  - Reconsider safety req.
  - Add/modify safety mech. (e.g. safe modes)
  - Modify architecture (e.g. redundancy)
  - Modify mapping
  - Succeeds :-) Safety leads to unsecure behaviour
  - Safety leads to degraded performance

Performance
- Succeeds :-)
- Fails
  - Reconsider performance req.
  - Reconsider algorithms
  - Modify architecture (Nb of cores, etc.)
  - Modify mapping
  - Performance issue due to safety mechanisms
  - Performance issue due to security mechanisms

Add/modify security mechanisms
Modify architecture (private bus, etc.)
Modify mapping
Reconsider security req.
Reconsider safety req.
Reconsider performance req.
SysML-Sec: SW Design

- Precise model of security mechanisms (security protocols)
- Proof of security properties: confidentiality, authenticity
- Channels between software blocks can be defined as private or public
  - This should be defined according to the hardware support defined during the partitioning phase
Case Studies

Cyber security of connected vehicles
- Safety/Security/Performance
- EVITA FP7 Partners: Continental, BMW, Bosch, ...
- VEDECOM

H2020 AQUAS
- Automated train sub-systems (ClearSy): Safety/Security/Performance
- Industrial Drives (Siemens): Safety/Security/Performance

Nokia
- Digital architectures for 5G networks (Safety/Performance)
Case Study: VEDECOM Autonomous Vehicle

Model

Verification

Tests
Constraints

- Standard: ISO26262
  - SOTIF: Safety Of The Intended Function
- Security: impact of potential attacks on safety
### Requirements

- **SecurityMain**
  - ID=0
  - Text="The autonomous system will be secure"
  - Kind="Functional"
  - Risk="Low"
  - Reference elements=""

- **ConfidentialGPS**
  - ID=8
  - Text="The system will not broadcast previous GPS locations"
  - Kind="Privacy"
  - Risk="Low"
  - Reference elements=""

- **ConfidentialKeys**
  - ID=7
  - Text="The system will ensure Confidentiality of Keys"
  - Kind="Confidentiality"
  - Risk="Low"
  - Reference elements=""

- **SensorTampering**
  - ID=11
  - Text="The system will verify sensor data"
  - Kind="Integrity"
  - Risk="Low"
  - Reference elements=""

- **AuthenticityFirmware**
  - ID=1
  - Text="The system will ensure authenticity of firmware"
  - Kind="Integrity"
  - Risk="Low"
  - Reference elements=""

- **ConfidentialityFirmware**
  - ID=0
  - Text="The system will ensure Confidentiality of firmware"
  - Kind="Confidentiality"
  - Risk="Low"
  - Reference elements=""

- **ConfidentialityReq**
  - ID=0
  - Text="The system will ensure Confidentiality"
  - Kind="Confidentiality"
  - Risk="Low"
  - Reference elements=""

- **AuthenticityFirmware**
  - ID=0
  - Text="The system will ensure authenticity of firmware"
  - Kind="Integrity"
  - Risk="Low"
  - Reference elements=""

- **V2XConfidentiality**
  - ID=2
  - Text="The system will ensure Confidentiality in the V2X system"
  - Kind="Confidentiality"
  - Risk="Low"
  - Reference elements=""

- **EthernetConfidentiality**
  - ID=3
  - Text="The system will ensure Confidentiality in the Ethernet network"
  - Kind="Confidentiality"
  - Risk="Low"
  - Reference elements=""

- **LANConfidentiality**
  - ID=0
  - Text="The system will allow data to be sent only in certain directions"
  - Kind="Controlled access (authorization)"
  - Risk="Low"
  - Reference elements=""

- **ConfidentialDataFlow**
  - ID=0
  - Text="The system will allow data to be sent only in certain directions"
  - Kind="Controlled access (authorization)"
  - Risk="Low"
  - Reference elements=""

- **FirmwareProtect**
  - ID=17
  - Text="Firmware will be encrypted"
  - Kind="Confidentiality"
  - Risk="Low"
  - Reference elements=""

- **(checkSumFirmware**
  - ID=19
  - Text="The system will use a checksum or something to ensure integrity of firmware"
  - Kind="Integrity"
  - Risk="Low"
  - Reference elements=""

- **V2XConfidentialitydata**
  - ID=21
  - Text="The system will only send traffic data over V2X"
  - Kind="Confidentiality"
  - Risk="Low"
  - Reference elements=""

- **networkFirmware**
  - ID=5
  - Text="The system will not send firmware on the network"
  - Kind="Confidentiality"
  - Risk="Low"
  - Reference elements=""

- **notAllSensors**
  - ID=0
  - Text="The attacker must not be able to compromise all sensors"
  - Kind="Integrity"
  - Risk="Low"
  - Reference elements=""
Attacks

- **Vehicle**
  - **root attack**: attackBraking
    - **OR**:
      - **AND**:
        - **attack**: preventObstacleDetection
        - **attack**: manipulateCamera
        - **attack**: manipulateLIDAR
      - **attack**: disableSensors
      - **attack**: preventDataComputation
      - **attack**: checkComponentStatus
    - **OR**:
      - **attack**: preventBrakingFunction
      - **attack**: preventBrakingCommandIssue
      - **attack**: preventBrakingCommand
    - **OR**:
      - **attack**: preventDataComputation
      - **attack**: disableSensors
      - **attack**: corruptControllerCode
      - **OR**:
        - **attack**: jamPerceptionCommunications
        - **attack**: forgeECUCommands
      - **OR**:
        - **attack**: forgePerceptionData
        - **attack**: jamECUCommunications
    - **OR**:
      - **attack**: authenticatePerceptionData
      - **attack**: authenticateECUCommands
      - **attack**: filterCommunications
      - **attack**: checkComponentStatus
Safety Verification (Before Mapping)

Reachability/Liveness

Queries

Safety Pragma
A[] Supervisor.running
Perception.distance<threshold -->
Supervisor.brakingOrder
Architecture and Mapping Views
Safety Verification (After Mapping)

Reachability Graph

Minimized RG
Security Verification

Dialog window

Backtracing

V2X_percData2

Satisfied Weak Authenticity:
PerceptionCalc1_encrypt_percData1_percData1 -> Supervisor.decrypt_percData1_dummy25
PerceptionCalc2_encrypt_percData2_percData2 -> Supervisor.decrypt_percData2_dummy32

Non Satisfied Authenticity:
PerceptionCalc1.signalstate_writechannel_Desig_sec_percStatus_percStatus_chData -> Supervisor
PerceptionCalc2.signalstate_writechannel_Desig_sec_percStatus2_percStatus2_chData -> Supervisor

Start  Stop  Close
Performance Verification

Latency

Bus/CPU Load

30/33 Sept. 2018 Institut Mines-Telecom
SW Design, Code generation, Test

- First SW model from mapping models
- SW model refinement
- SW model verification (safety, security)
- Code generation
  - (Virtual) Prototyping, test
Conclusion and Future Work

Achievements: SysML-Sec

- Methodology for designing safe and secure embedded systems
- Fully supported by TTool
- Applied to different domains, e.g., automotive systems, IoTs, malware

Future work

- Security risk assistance and backtracing
- Assistance to handle conflicts between security/safety/performance
  - Design space exploration
To Go Further ...

Web sites


References
