Model-Driven Engineering for Safety, Security and Performance:
SysML-Sec

Ludovic APVRILLE
ludovic.apvrille@telecom-paristech.fr

InS3PECT’2017
Outline

Context: Security for Embedded Systems
Embedded systems

SysML-Sec
Method
SysML-Sec

Case study and Demo
Case Study and Demo

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

(C) Wired - ABC News

(C) Hospira
Examples of Threats (Cont.)

Internet of Things

- Proof of concept of attack on IZON camera [Stanislav 2013]

- Vulnerability on fitbit [Apvrille 2015]

- Hacking a professional drone [Rodday 2016]
Finding Vulnerabilities onIoTs

What’s inside? Let’s look together!
Inside a Fitbit

Don’t try this at home!
Inside a Fitbit (Cont.)

Again: don’t try this at home!
Inside a Fitbit (Cont.)
Fitbit: Hardware Components

- Vibrator
- LIPo battery
- STM LIS2DH - Triaxial MEMS
- STM 32L151CB
- NRF 8001 1386KV
- TI Charger BQ24040
- LEDs
Then, How to Identify Vulnerabilities?

Investigations

- JTAG interface
- Testing ports
- Firmware analysis
- Memory dump
- ...

You want to better resist this?

Develop your system with security in mind from the very beginning

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

Main idea

- **Holistic approach**: bring together experts in embedded systems, system architects, system designers and security experts

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
SysML-Sec: Methodology

Requirements

Functional view
Architectural view
Mapping view
SW/HW Partitioning

Use case view
Scenario view
SW Analysis

Structural view
Behavioral view
SW Design

Simulation Formal analysis

Assumptions

Attacks

Simulation Formal analysis

Test

Deployment view

Fully supported by TTool

12/29 Dec. 2017 Institut Mines-Telecom

SysML-Sec

Case study and Demo

Conclusion
Partitioning

Before mapping

▶ Security mechanisms can be captured but not verified

After mapping

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

Modeling

Automatic Verification

Safety

Performance

Security
Safety and Security Mechanisms

Data Encryption/Authentication

Safety

Security

Performance

ECUCommand(1)

chl

SE

ECUCommand(1)
Safety and Security Mechanisms

Redundancy/Coherence Check

Add security

Add security

Safety

Security

Performance
Safety and Security Mechanisms

Failsafe mode

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

Safety

Security

Performance
## Partitioning Approach

### Requirements
- **Security**
  - Automated generation
- **Safety**
- **Performance**

### System design

### Verification of design w.r.t. requirements

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

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

#### Performance
- Succeeds ☺️
  - 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

---

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

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

Performance issue due to safety mechanisms
Reconsider algorithms
Modify architecture (Nb of cores, etc.)
Modify mapping

---

19/29 Dec. 2017
Institut Mines-Telecom
SysML-Sec
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 Study: Autonomous Vehicle

Verification

Model
Safety Verification (Before Mapping)

Reachability/Liveness

Queries

Safety Pragma
A[] Supervisor.running
Perception.distance<threshold -->
Supervisor.brakingOrder
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_dummy2
PerceptionCalc2_encrypt_percData2_percData2 ==> Supervisor.decrypt_percData2_dummy3

Non Satisfied Authenticity:
PerceptionCalc1.signalstate_writechannel_Design_sec_percStatus.percStatus_chData ==> S
PerceptionCalc2.signalstate_writechannel_Design_sec_percStatus2.percStatus2_chData ==> S

Start Stop Close
Performance Verification

Latency

Bus/CPU Load
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
- Improve security provers
- Assistance to handle conflicts between security/safety/performance
  - Design space exploration
To Go Further ...

Web sites

▶ https://ttool.telecom-paristech.fr

References