



Institut  
Mines-Telecom

## **SysML-Sec Attack Graphs: Compact Representations for Complex Attacks**

Ludovic Apvrille  
[ludovic.apvrille@telecom-paristech.fr](mailto:ludovic.apvrille@telecom-paristech.fr)

Yves Roudier  
[yves.roudier@eurecom.fr](mailto:yves.roudier@eurecom.fr)

GraMSec'2015





# Outline

## Context: Security for Embedded Systems

Embedded systems

SysML-Sec

Attack trees

Contribution

Conclusion



# Designing Safe and Secure Embedded Systems: SysML-Sec

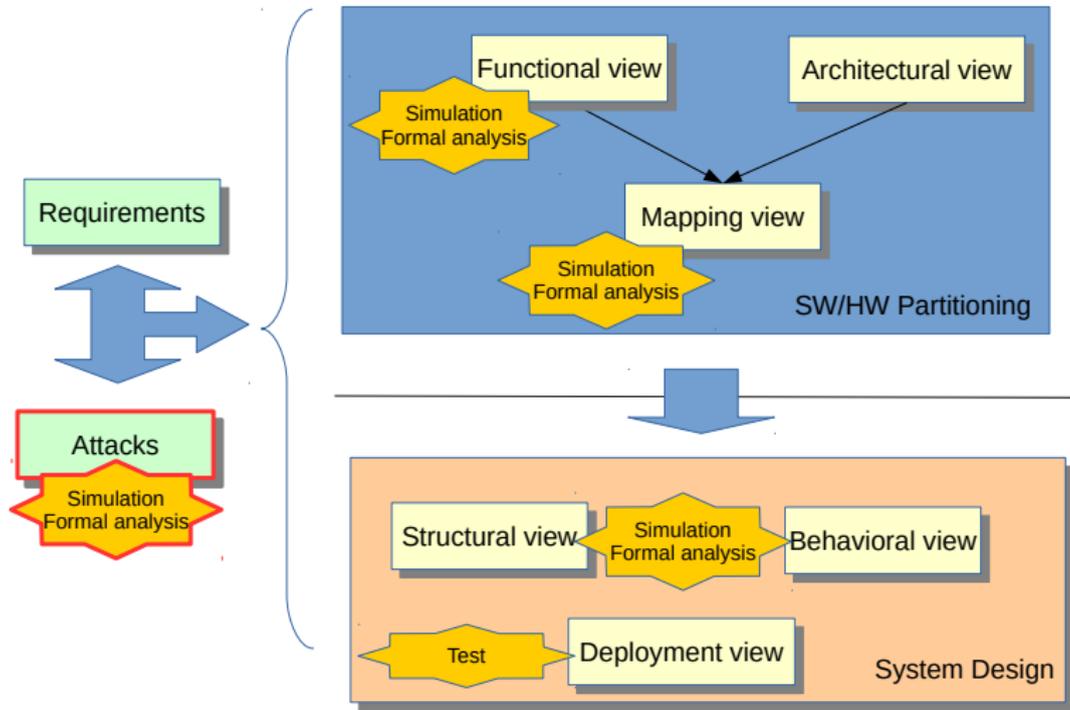
## Main idea

- ▶ **Holistic approach:** bring together experts in embedded 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



Fully supported by TTool



# Outline

Context: Security for Embedded Systems

Attack trees

Attack trees

Contribution

Conclusion

# Google-izing Attack Trees

The collage contains several diagrams illustrating attack trees and related concepts:

- Steal customer data:** A tree starting with 'Steal customer data' and branching into 'Steal login', 'Steal password', and 'Steal session ID'. 'Steal login' further branches into 'Steal user name', 'Steal user ID', 'Steal user email', and 'Steal user phone number'. 'Steal password' branches into 'Steal password in clear text', 'Steal password in hashed form', and 'Steal password in encrypted form'. 'Steal session ID' branches into 'Steal session ID in clear text', 'Steal session ID in hashed form', and 'Steal session ID in encrypted form'.
- Threat #1: Changing authentication credentials over the network:** A tree starting with 'Threat #1' and branching into '1.1 Check for credentials, split over the network' and '1.2 Attacker uses network monitoring tools'. '1.2' further branches into '1.2.1 Attacker recognizes credential data' and '1.2.2 Attacker recognizes credential data'.
- Attack Tree Theory:** A diagram with a blue background showing a tree structure with nodes labeled 'Attack Tree Theory' and 'Attack Tree Theory'.
- Attack Tree:** A diagram showing a tree structure with nodes labeled 'Attack Tree' and 'Attack Tree'.
- Attack Tree Online Banking Applications:** A diagram showing a tree structure with nodes labeled 'Attack Tree Online Banking Applications' and 'Attack Tree Online Banking Applications'.
- Attack Tree Theory (Blue):** A diagram with a blue background showing a tree structure with nodes labeled 'Attack Tree Theory' and 'Attack Tree Theory'.
- Attack Tree Theory (Yellow):** A diagram with a yellow background showing a tree structure with nodes labeled 'Attack Tree Theory' and 'Attack Tree Theory'.
- Attack Tree Theory (Green):** A diagram with a green background showing a tree structure with nodes labeled 'Attack Tree Theory' and 'Attack Tree Theory'.
- Attack Tree Theory (Red):** A diagram with a red background showing a tree structure with nodes labeled 'Attack Tree Theory' and 'Attack Tree Theory'.
- Attack Tree Theory (Purple):** A diagram with a purple background showing a tree structure with nodes labeled 'Attack Tree Theory' and 'Attack Tree Theory'.
- Attack Tree Theory (Orange):** A diagram with an orange background showing a tree structure with nodes labeled 'Attack Tree Theory' and 'Attack Tree Theory'.
- Attack Tree Theory (Pink):** A diagram with a pink background showing a tree structure with nodes labeled 'Attack Tree Theory' and 'Attack Tree Theory'.
- Attack Tree Theory (Brown):** A diagram with a brown background showing a tree structure with nodes labeled 'Attack Tree Theory' and 'Attack Tree Theory'.
- Attack Tree Theory (Grey):** A diagram with a grey background showing a tree structure with nodes labeled 'Attack Tree Theory' and 'Attack Tree Theory'.
- Attack Tree Theory (White):** A diagram with a white background showing a tree structure with nodes labeled 'Attack Tree Theory' and 'Attack Tree Theory'.
- Attack Tree Theory (Black):** A diagram with a black background showing a tree structure with nodes labeled 'Attack Tree Theory' and 'Attack Tree Theory'.

# Attack Trees

## Definition and purpose

- ▶ Originate from fault trees, introduced by Bruce Schneier (1999)
- ▶ Depict how a system element can be attacked
  - ▶ Helps finding attack countermeasures
- ▶ Root attack, children, leaves
- ▶ OR and AND relations between children



# Attack Trees: Related Work

- ▶ Generation of ATs from other formalisms [Vigo 2014]
- ▶ Semantics extensions
  - ▶ [Khand 2009]
    - ▶ *PAND*, *k-out-of-n*, *CSUB*, *SEQ*, ...
  - ▶ [Zhao 2014]
    - ▶ Permissions and capabilities on nodes
    - ▶ Applied to malware analysis
- ▶ Security assessment
  - ▶ Privilege graphs [Dacier 1996]
  - ▶ Petri nets [Dalton 2006] [Pudar 2009]
  - ▶ Markov processes [Piètre-Cambacédès 2010]

# Attack Trees: A Few Issues

## Semantics

- ▶ Semantics of AND and OR is limited to express complex attack scenarios
  - ▶ No ordering between attacks
  - ▶ No temporal operators

## Relation with other development stages

- ▶ No relation with (security) requirements
  - ▶ More generally, not integrated into methodologies
- ▶ No relation between attacks and the HW/SW components of the system
  - ▶ Difficult to figure out the where and which of countermeasures



# Outline

Context: Security for Embedded Systems

Attack trees

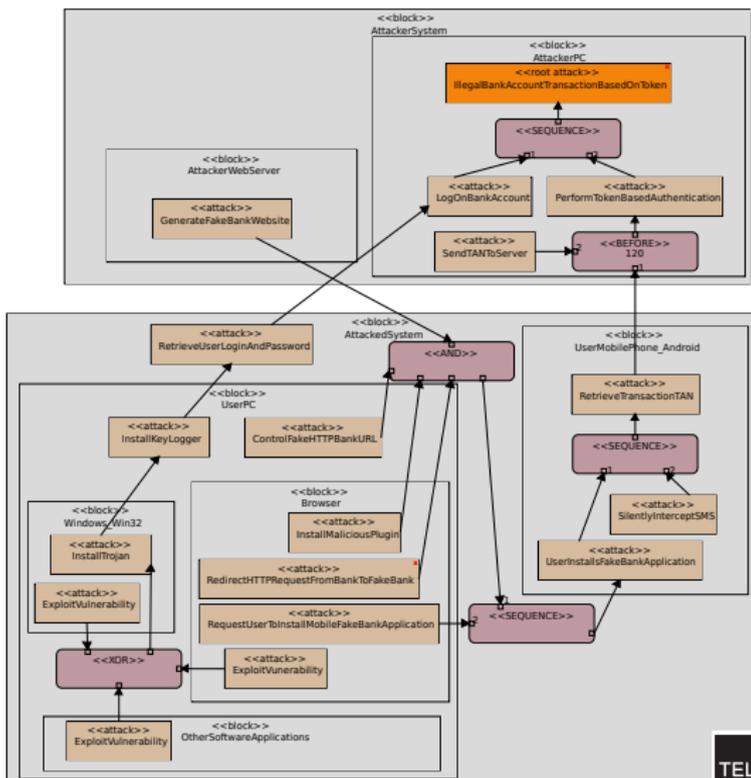
**Contribution**

New operators

Conclusion

# Overview (with an Example)

- ▶ SysML Parametric diagram
- ▶ Asset = Block
- ▶ Attacks = Attributes of blocks
- ▶ Relation between attacks = Constraints
- ▶ Formal semantics
  - ▶ Timed automata



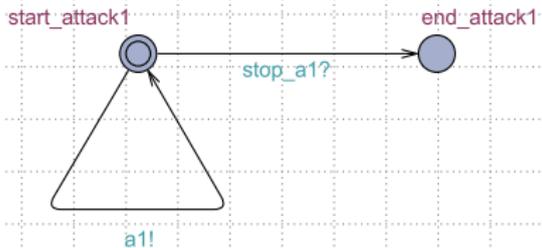


# Semantics

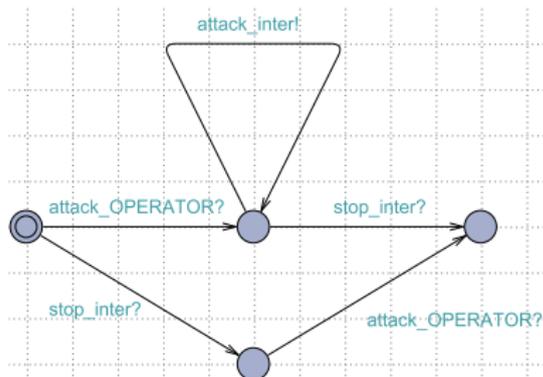
- ▶ Attacks
- ▶ Intermediate attacks
- ▶ Root attack
- ▶ Constraints
  - ▶ AND, OR, XOR, SEQUENCE, BEFORE, AFTER

# Semantics of Attacks

## Attack

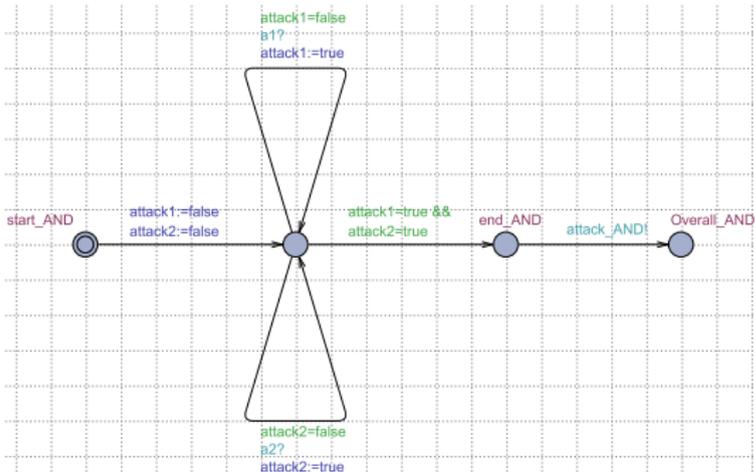


## Intermediate Attack



# Semantics of Constraints

## AND

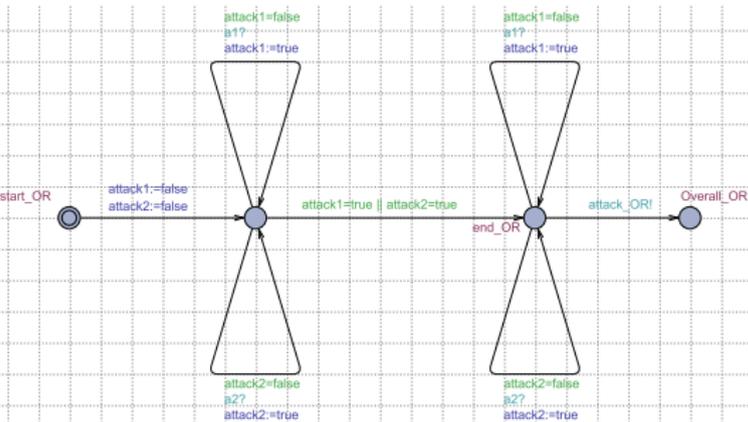


## SEQUENCE

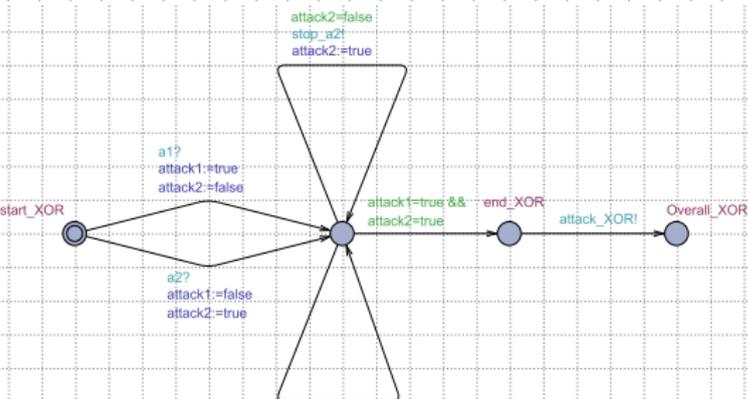


# Semantics of Constraints (Cont.)

OR

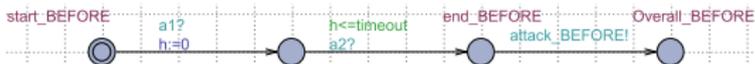


XOR



# Semantics of Constraints (Cont.)

BEFORE

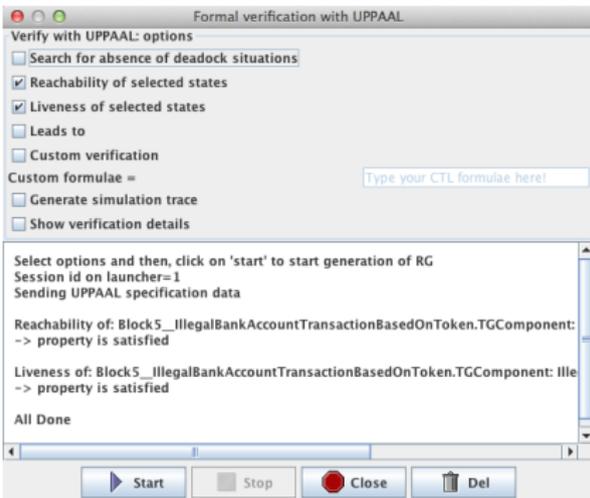


AFTER



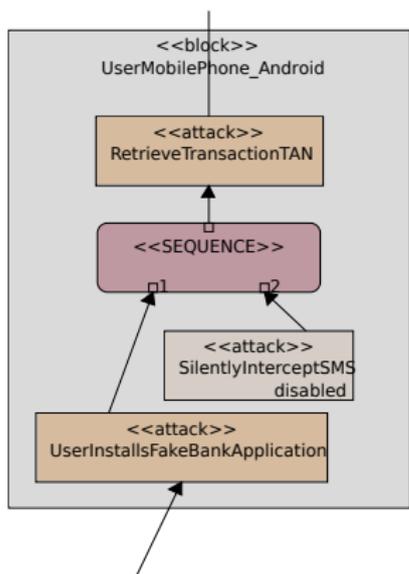
# Formal Verification

- ▶ Reachability of an attack  $a$
- ▶ Liveness of an attack  $a$
- ▶  $a_1$  Leads to  $a_2$  ( $a_1 \rightsquigarrow a_2$ )



# Disabling Attacks

- ▶ Right click to disable/enable an attack



Formal verification with UPPAAL

Verify with UPPAAL: options

- Search for absence of deadlock situations
- Reachability of selected states
- Liveness of selected states
- Leads to
- Custom verification

Custom formulae =

- Generate simulation trace
- Show verification details

Select options and then, click on 'start' to start generation of RG  
 Session id on launcher=1  
 Sending UPPAAL specification data

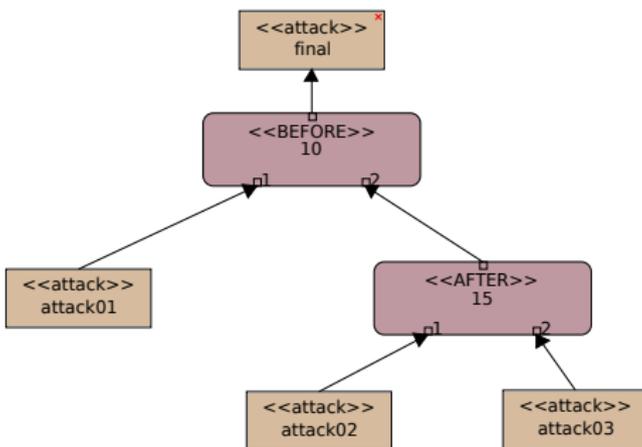
Reachability of: Block5\_IllegalBankAccountTransactionBasedOnToken.TGComponent:  
 -> property is NOT satisfied

All Done

Start Stop Close Del

# Temporal Compatibility

- ▶ Temporal constraints may impact attacks reachability/liveness





# Outline

Context: Security for Embedded Systems

Attack trees

Contribution

**Conclusion**

Conclusion, future work and references

# Conclusion and Future Work

## Achievements

- ▶ Extended and formally defined attack trees
- ▶ Integrated into SysML-Sec
- ▶ Fully supported by TTool
- ▶ Applied to different domains, e.g., malware, automotive systems

## Future work

- ▶ Handling new situations
  - ▶ Cycles, nb of iterations, priorities
- ▶ Quantitative assessments of threats

## To Go Further ...

### Web sites

- ▶ <https://sysml-sec.telecom-paristech.fr>
- ▶ <https://ttool.telecom-paristech.fr>

### References (SysML-Sec)

- ▶ Ludovic Aprville, Yves Roudier, "SysML-Sec: A SysML Environment for the Design and Development of Secure Embedded Systems", Proceedings of the INCOSE/APCOSEC 2013 Conference on system engineering, Yokohama, Japan, September 8-11, 2013.