IT Security in Automotive Software Development

***<u>Sandro Schulze</u>*, Mario Pukall, Tobias Hoppe**

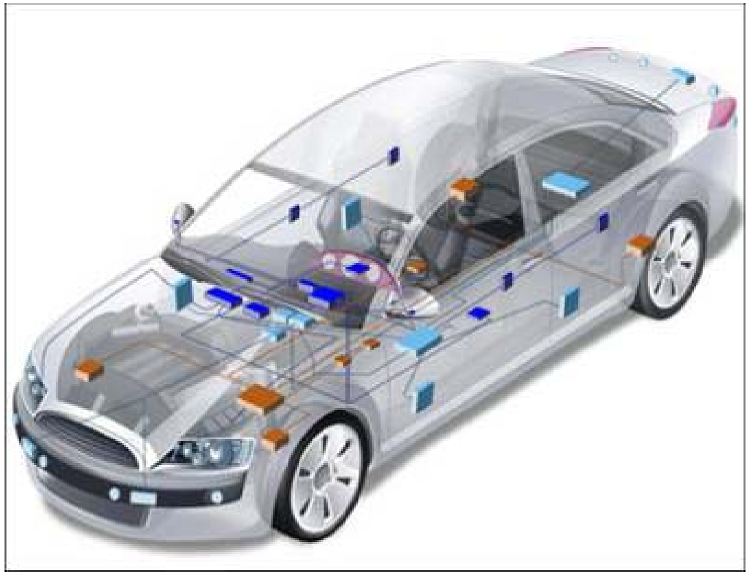
OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG
INF FAKULTÄT FÜR INFORMATIK

Outline

- Automotive Systems
- Motivation
  - IT Security
  - Software Development
- Specifying Security Requirements
  - Modeling Functional Dependencies
  - Formalizing the Propagation of Security Requirements
- Conclusion

Automotive System

- **Consists of a multitude of embedded systems (ECUs)**
- **Constraints with respect to resources, e.g., memory, processor etc.**
- **Hard real-time requirements in the dimension of ~ 10 ms**
- **Communication via bus protocols (CAN, LIN, MoST) → no authentification**
- **Functionality realized by software**



Motivation – IT Security

- **Automotive system is similar to networked IT system → similar problems like in desktop-IT systems**
- **Additional vulnerabilities caused by new technologies (e.g., Car-2-Car)**
- **Attacks are possible**
  - **From *inside*, e.g., CDs, wireless communication**
  - **From *outside*, e.g., by adding new devices**
- **Mostly, attacks aim at manipulating the software**

→ **IT security has to be considered during SW development process**

Motivation – Software Development

- **Demand for efficient SW development**
  - **Decreasing development costs**
  - **Decreasing the complexity of the system**
- **Different approaches exist**
  - **Software Engineering concepts, e.g., software product lines (SPLs)**
  - **Model driven development**
  - **Requirements Engineering**
- **IT security not considered in this context → retrofitted code for known vulnerabilities**
- **Increases complexity and risk of IT security attacks**

Specifying Security Requirements

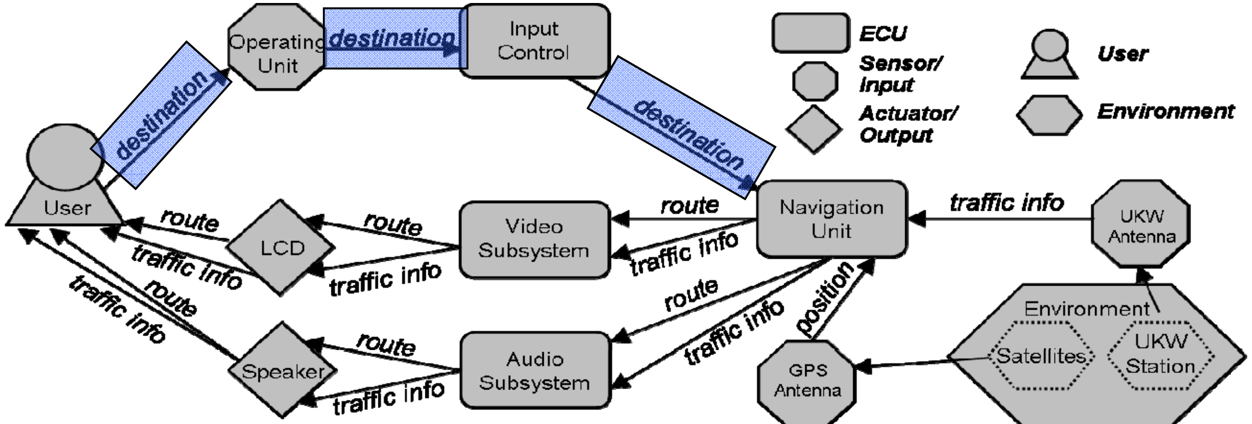
- **Idea: IT security in early stages of SW development**
- **Approach divided into two parts**
  - **Modeling functional dependencies**
  - **Formalizing the propagation of IT security requirements**
- **Objective: security requirements in requirements engineering (RE) stage**
  - **Early specification → usage during design and implementation**
  - **Model-based approach → integrated into systems engineering process**

Modeling Functional Dependencies (1)

- **Automotive HW (e.g., ECUs) take part in several functionalities → dependencies between several devices**
- ***Logical view* on AS enables investigation of functional dependencies**
  - **Different degrees of granularity**
  - **Possible vulnerabilities can be detected “visually”**
- ***Function Nets (FN)* as modeling approach**
  - **Reduces modeling complexity**
  - **Focus on functional/logical view**
- **Supported by standard modeling languages, e.g., SysML**

Modeling Functional Dependencies (2)

- **Example**



Confidentiality *(C)*
Integrity *(I)*
Availability *(A)*
Authenticity *(U)*
Non-Repudiability *(N)*
Privacy *(P)*

- **Abstract model representation with nodes and data flow**
- **Can be divided into *graphs,* representing certain features**
- **Security requirements depend on**
  - **Data item (characterizing the graph)**
  - **Node type *(Consumer* or *Provider)***

Formalization

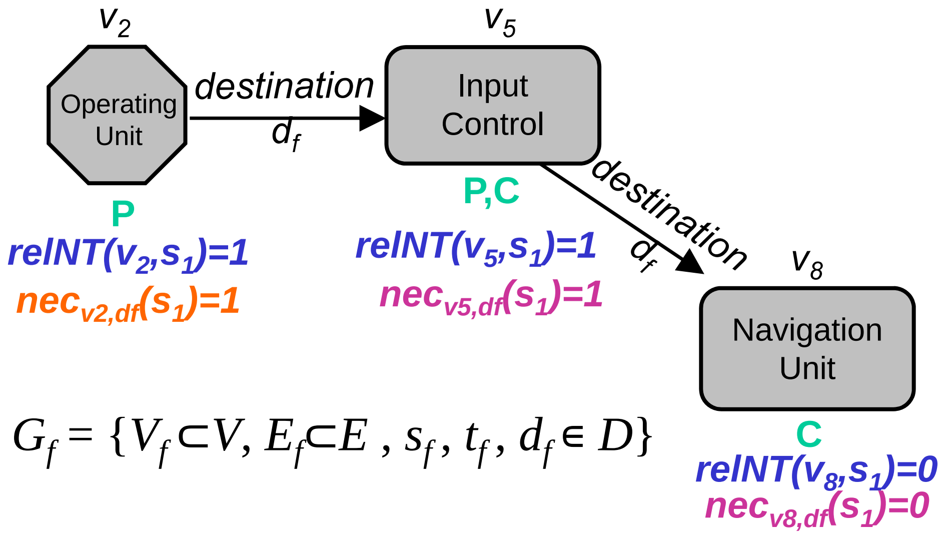
- **Target: specifying security requirements for certain components based on data items**
- **Idea: enhancing function nets by security requirements**
- **Exploiting functional dependencies for propagation of security requirements**
- **Approach:**
  - **Dividing function net into graphs (data-driven)**
  - **Determine security requirements for single graphs (using formalization)**
  - **Composing graphs to function net**
  - **Evaluating security requirements for the whole automotive system**

Formalization – Basis Definitions

- **Set of all *vertices* (nodes), e.g., ECUs:** $V = \{v_1, \ldots v_{nv}\}$
- **Set of all *data items*:** $D = \{d_1, \ldots d_{nd}\}$
- **Set of all *edges*:** $E = \{e_1, \ldots e_{ne}\}$ **where**
  - **A triple** $e_m = \{v_j, v_k, d_m\}$ **describes a certain edge with** d **as exchanged data**
  - $\{v_j, v_k\} \in E$ → **a pair of connected vertices**
- **Set of all *graphs*:** $G = \{G_1, \ldots G_{ng}\}$ **with**
  - $G_m = \{V_m \subset V, E_m \subset E, s_m, t_m, d_m \in D\}$ **as a certain graph**
  - **Functions** $s_m, t_m: E \mapsto V$ **assigning source/target vertices to the respective edges**
- **Set of all *function nets*:** $F = \{f_1, \ldots, f_{nf}\}$ **with** $f_i = \{G_i \subset G\}$ **as a particular one**
- **Set of all security aspects:** $S = \{C, I, A, U, N, P\}$
  - **Consumer** → $CR \subset S = S / \{C, P\}$
  - **Provider** → $PR \subset S = S / \{I, A, U\}$

Formalization – Propagation of Security Requirements

- **Precondition → function nets for whole AS available**
- **Objective: specifying security requirements of a whole graph by exploiting functional dependencies**
- **Example: graph for the data item "destination" (d_f)**



$$G_f = \{V_f \subset V,\ E_f \subset E\ ,\ s_f,\ t_f,\ d_f \in D\}$$

The graph provides:
- ***node types*** (*P*=Provider, *C*=Consumer) for the respective data
- ***Relevance*** of the security aspect $relNT(v_i,s_j)$
  - ⇨ $s_j \in CR$ or $s_j \in PR$?

Further steps:
- Determining **necessary security requirements** for initial node via $nec_{vi,dj}(s_k)$
- Iterative **propagation** of security aspects for the data item on further nodes via $delegate(v_n,v_m,s_i)$

- **Afterwards: composing graphs to function net**
- **Vertices/components responsible for countermeasures?**
  - **⇨Future research**

Conclusion/Future Work

- **Automotive systems exhibit vulnerabilities regarding (software) manipulation → importance of IT security**
- **To be considered *early* in software development process → often neglected**
- **(model-based) approach for specifying security requirements**

- **Evaluating the approach in the context of domain-specific models (e.g., Simulink)**
- **Usage of sophisticated graph concepts (*attributed, typed*)**
- **Managing composed data**
- **Draw conclusions regarding suitable countermeasures**

Thank you !

Project page: http://omen.cs.uni-magdeburg.de/automotive/cms/

Questions? Notes? Advices?

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG
INF FAKULTÄT FÜR INFORMATIK