Secure Vehicle Communication

SEVECOM

In-vehicle Intrusion Detection System

Daimler AG

Albert Held

Michael Müter

In-Vehicle Intrusion Detection

SEVECOM

- Besides preventive measures, monitoring of the vehicle's system to detect possible attacks is necessary
- Investigation if state-of-the art IDS technologies can be adapted for vehicles
  - Work started March 07

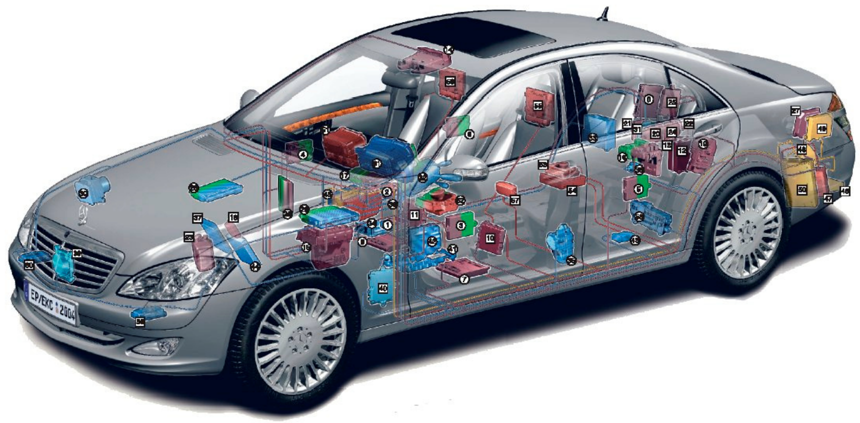
- Overview:
  - Challenges for future cars
  - Characteristics of scenarios
  - Approach for IDS
  - Example: Formal model
  - Summary

Future Cars

- Wireless interfaces
- Internal harddisk
- Additional interfaces (CD, DVD, USB, ...)
- Integration of consumer devices
- DRM
- Software based and remote functions

  => open system

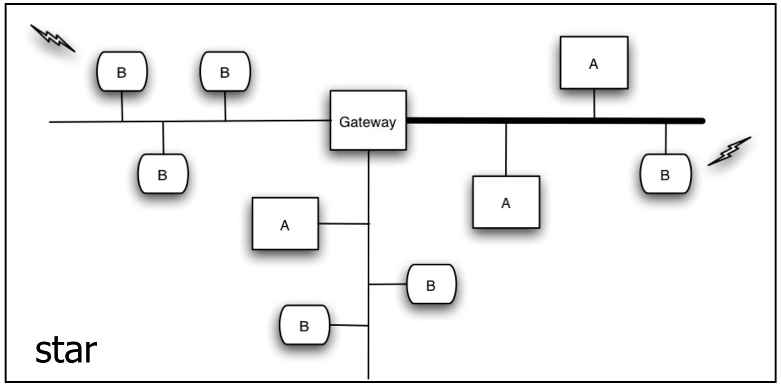
- Increased risks by hackers, malware, ...
  - Protocol attacks by hostile devices, sniffers, viruses, ...

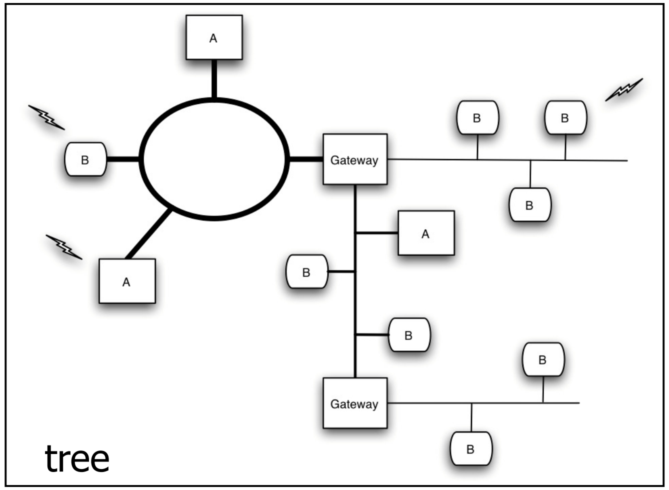


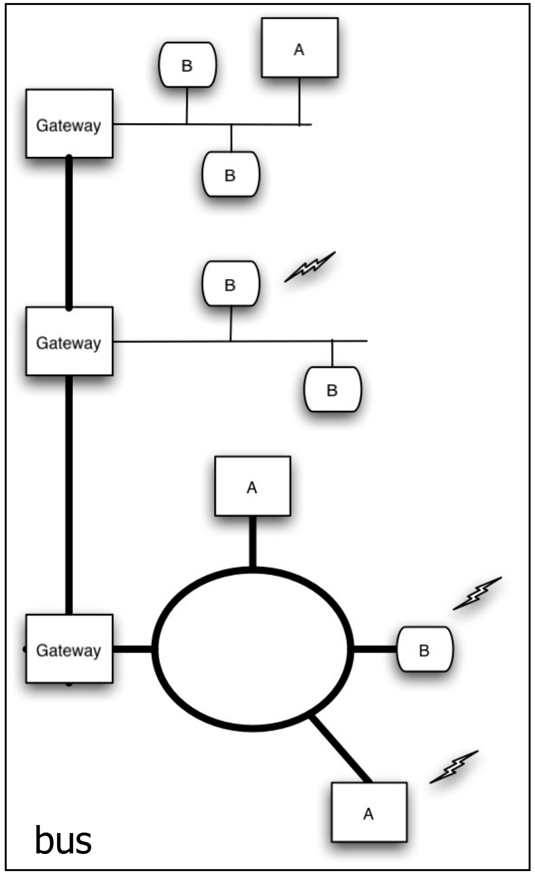
Characteristics of scenario

- Different types of „systems“: entry-level - high-level
- Different types of networks
  - Topology, Data rate, Access mechanisms, ...
- Different types of devices
  - Embedded devices - „PC-like“ devices
- Multiple access points
  - Wireless communication, Interfaces
- Multiple operation modes
  - E.g. diagnosis, ...
  - System is restarted every x hours
- Performance constraints
  - Real-time requirements but limited performance devices
- Autonomous operation
  - Vehicle should work independently from driver

In-Vehicle Scenarios





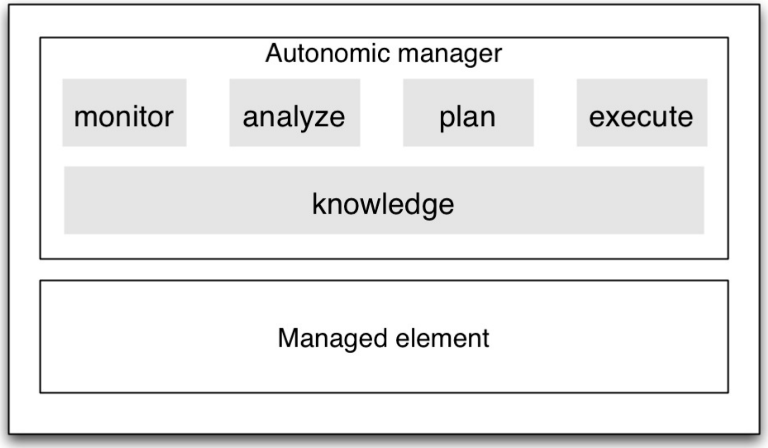


Approach

- Evaluation of existing IDS
  - Knowledge based   only known attacks, frequent updates necessary
  - Anomaly detection specification of system's behaviour

  => Anomaly detection

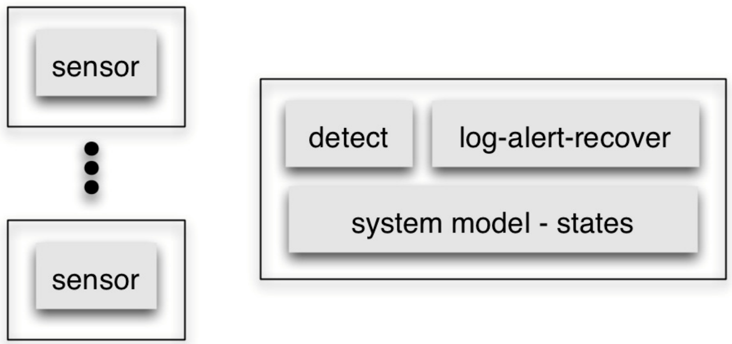
- Evaluation of Autonomous Computing concepts
  - General Model: IBM autonomic element



In-Vehicle IDS

- Identification of requirements
  - Formal model to describe the system's security state
  - Identification of secure/insecure states
  - Monitoring (sensors)
  - Detection engine
  - Logging
  - Alerting
  - Recovery measures

- Specification of the components of the in-vehicle IDS



Example Formalization

Towards a formal model of intrusion detection:

- The Distributed Computer System (DCS) consists of a set of communicating objects
- Set of DCS objects comprises active and passive objects: $\Re = \Re_A \cup \Re_P$
- Each object $r \in \Re$ provides
  - Access rights A
    (r_i, r_j) $r_i \in \Re_A, r_j \in \Re$
  - Workload function $\lambda : \Re \rightarrow [0..1]$
    $\lambda(r)$ is the current workload for an object $r \in \Re$
- Complement of further object attributes (memory,…)

Example Formalization

Towards a formal model of intrusion detection:

- Object state for passive objects

a) $$S_{r_P} = (In(r), L(\lambda(r_P)))$$

b) $$In(r_P) = \{r \in \Re_A \mid \exists a \in A_{r_P} : r \xrightarrow{a} r_P\}$$

c) $$L(\lambda(r_P)) = [0,1]$$

0 : Workload <= limit
1 : Workload > limit

- Analog definition for active objects
- Distributed Computer Systems (DCS): $\Theta = < \Re, \{S_r \mid r \in \Re\} >$
- Object state *unsafe*, if

$$L(r) = 1 \quad or \quad \exists r' : r' \xrightarrow{*} r \wedge (r', r) \notin A_r$$

=>DCS system *unsafe* if at least one object in unsafe state

Summary and next steps

- Summary:
  - Complex vehicle architecture and various scenarios
  - Necessity to protect vehicle system against attacks
  - Consideration of two different approaches
- Next steps:
  - Work on formal model and security states
  - Definition of basic sensors
  - Simulation tool to check the feasibility of the approach