VECTOR

Automotive Cybersecurity - Challenges and Practical Guidance

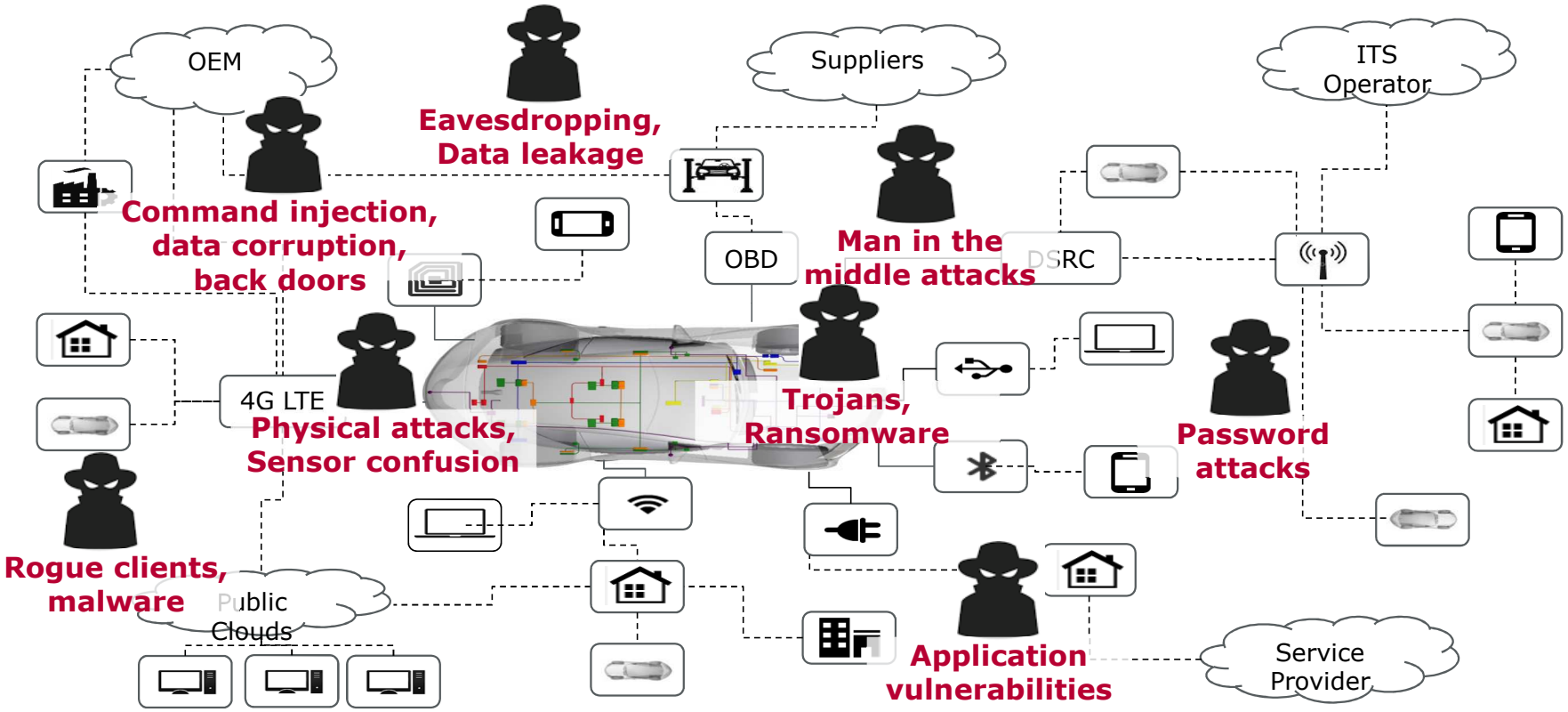
Vector Technology Days 2019 - 2019-10-23, Böblingen, Germany

V1.0 | 2019-10-23

Agenda

1. Introduction
2. Threat Analysis and Risk Assessment
3. Design for Security
4. Security Testing
5. Summary

ACES (Autonomy, Connectivity, Efficiency, Services) ▶ Cyberattacks



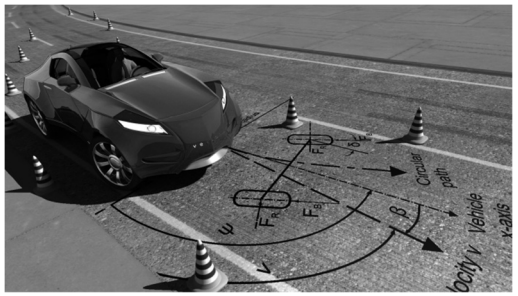
Security will be the major liability risk in the future.
Average security breach is detected in 70% of all cases by third party – after 8 months.

Automotive Trends Impact Safety and Security

1. Powertrain

Energy efficiency

Unintended speed change



2. Driver Assistance

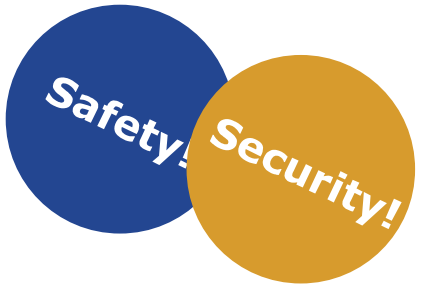
Autonomous driving

Signal confusion

3. Connectivity

Always connected

Sudden Driver distraction



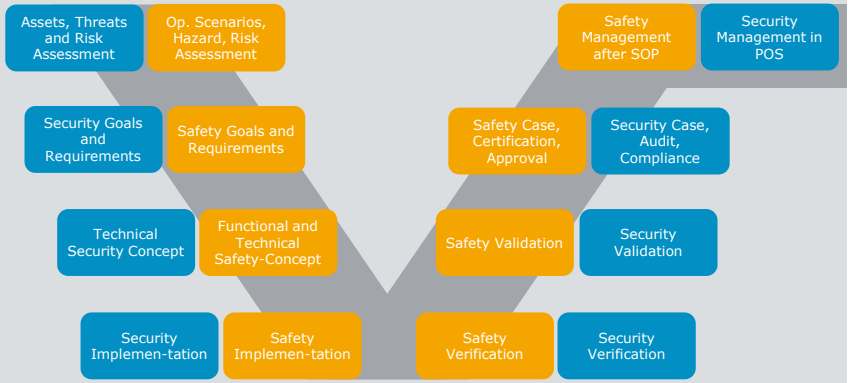
VECTOR

Security and Safety Standards Evolve in Parallel

Functional Safety
(IEC 61508, ISO 26262, ISO/PAS 21448)

- Hazard and risk analysis
- Functions and risk mitigation
- Safety engineering

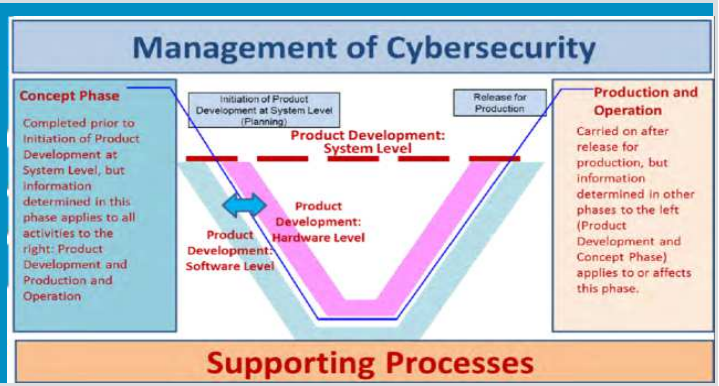
ISO 26262:2018 does not address security, but requires trade-offs without impact on functional safety.



+ Security
(ISO 27001, ISO 15408, ISO 21434, SAE J3061)

- Threat analysis and risk assessment
- Abuse, misuse, confuse cases
- Security engineering

Security and Safety are inter-related and demand holistic systems engineering



For (re) liable and trusted operation **security is a enabler for safety**

Upcoming ISO/SAE 21434 Standard for Automotive Cybersecurity

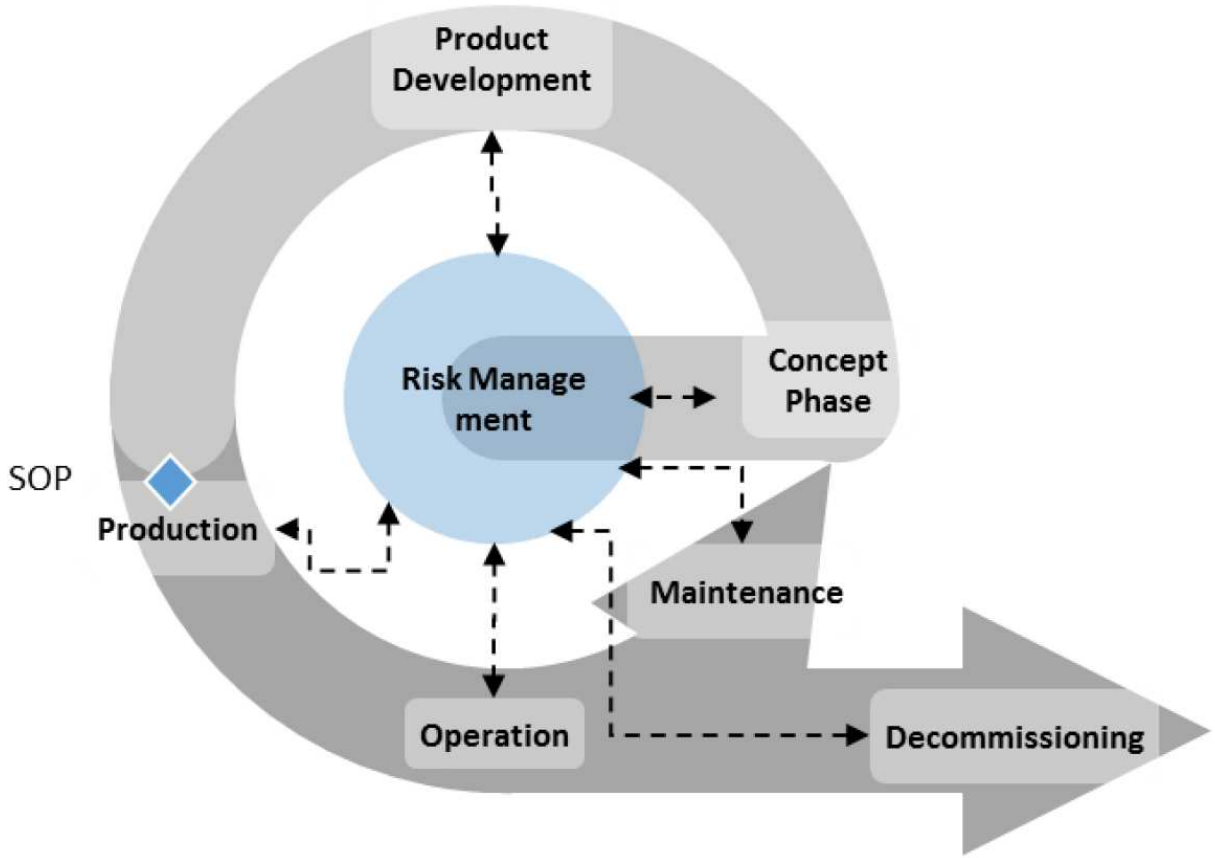
Planning

- Kickoff - 17.10.2016
- Currently: Committee Draft
- Release: 2020 (Planned)

Approach

Risk-oriented approach following the Vector method for the whole lifecycle:

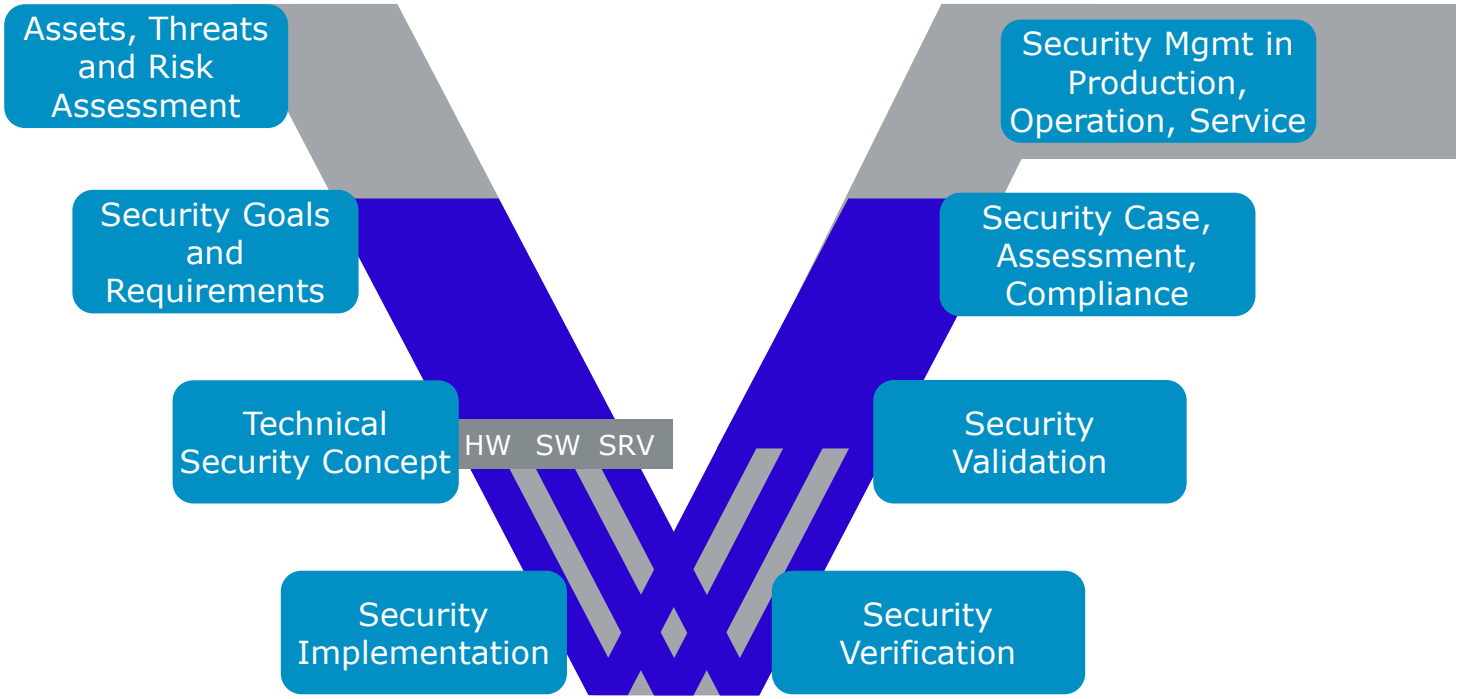
- Concept/design phase
- Product development
- Production (roll out)
- Operation
- Decommissioning (roll over)



Focus on governance - ISO 21434 does NOT mandate technologies or solutions

VECTOR

Security Engineering



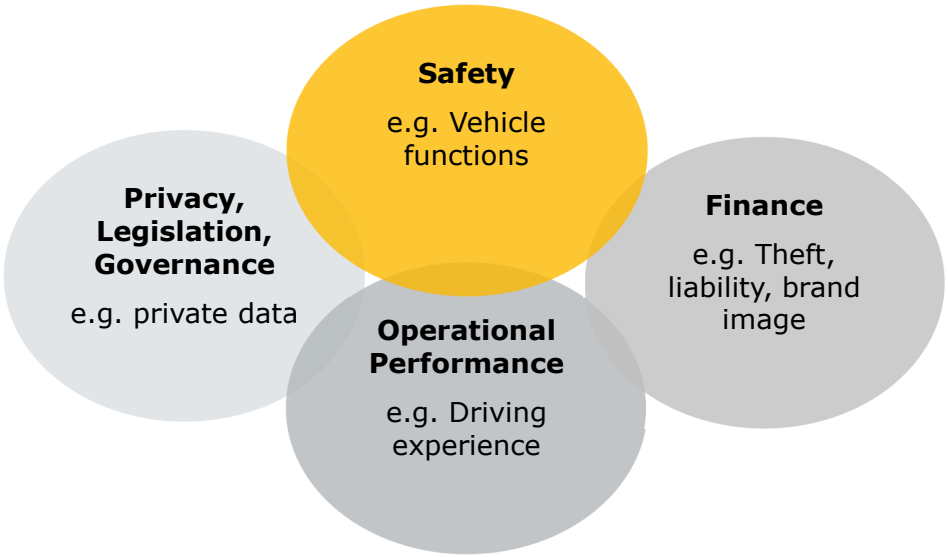
Most security attacks are process and implementation related.
They rarely lie within the cryptographic protocols and algorithms.

Agenda

1. Introduction
2. Threat Analysis and Risk Assessment
3. Design for Security
4. Security Testing
5. Summary

TARA - Identify and Agree on Assets

Specific automotive asset categories

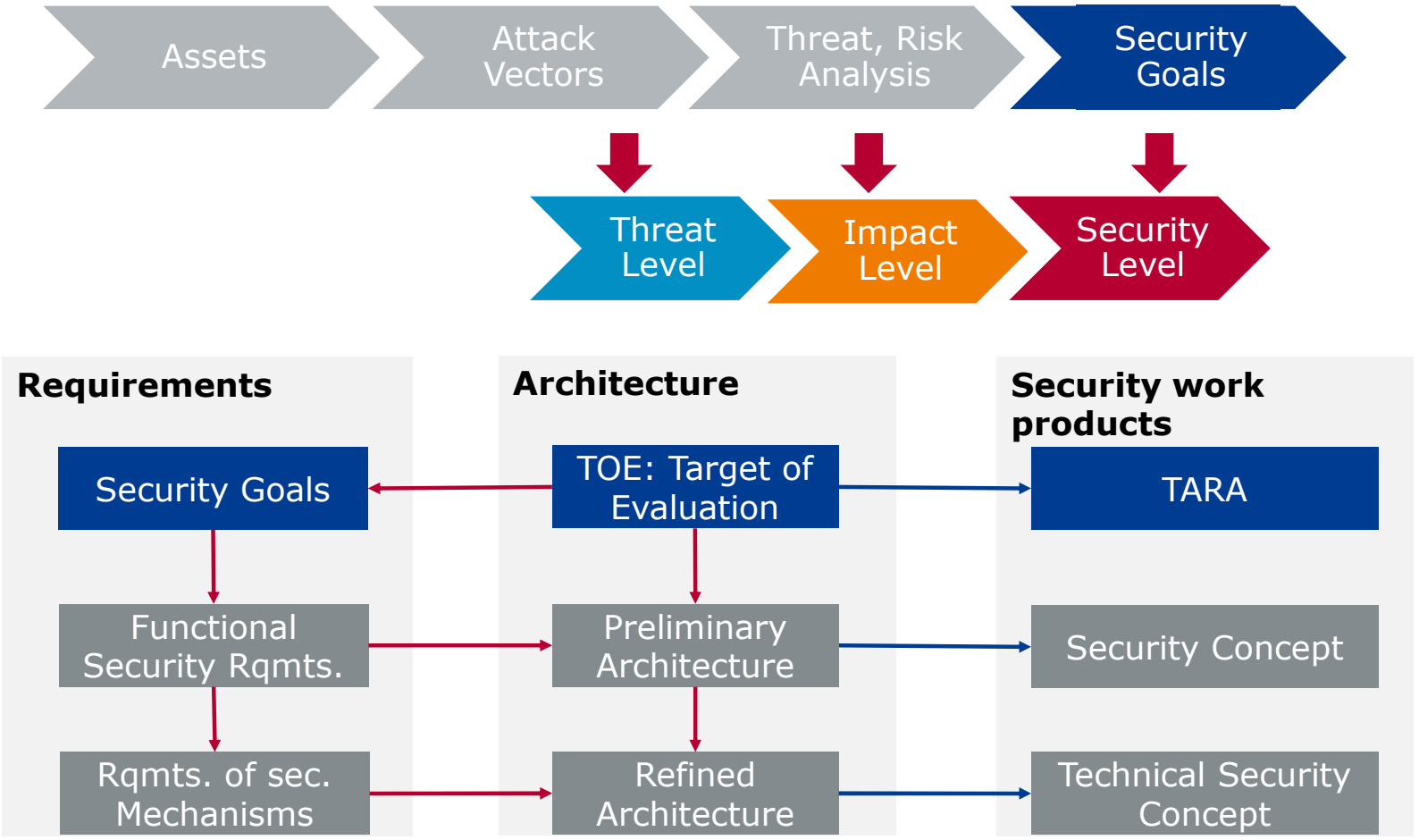


Checklist to identify assets

- Which information, algorithm or intellectual property shall **remain confidential?**
- Which data (e.g. configuration parameters) shall **remain unchanged?**
- Which functions or procedures shall be **exclusively applied** by e.g. the OEM?
- Which functions or data shall be always **available?**
- Which company **guidelines** or **legal requirements** on data or procedures must be fulfilled?

Automotive Cybersecurity focuses on Confidentiality, Integrity, Authenticity, Availability, Governance (CIAAG)

Identification of Security Goals by Threat Analysis and Risk Assessment (TARA)



VECTOR

Determine Necessary Security Level with TARA Results

Asset ID	Asset / Vehicle Function	CIAAG	Attack vector	Potential effect of attack	Threat ID	Threat	Expertise	Expertise numerical	Window of Opportunity	WoO numerical	Equipment / Effort	Effort numerical	Threat numerical	Threat level (high=4; low=1)	Safety	Financial	Operational	Privacy	Impact Level	SGID
Ast 01	Safety-Mechanisms	Avail	Availability: Attacker floods CAN-Bus and thereby tries to disable vehicle primary functions.	Attacker disables engine control during an overtaking maneuver if system can impact safety-critical functions.	Tht-1	Not further considered on advice of client because the HU is rated QM with respect to ISO 26262.	Layman	0	Critical	0	Standard	0	0	4	No injury	No impact	No impact	No effect	No impact	n/a

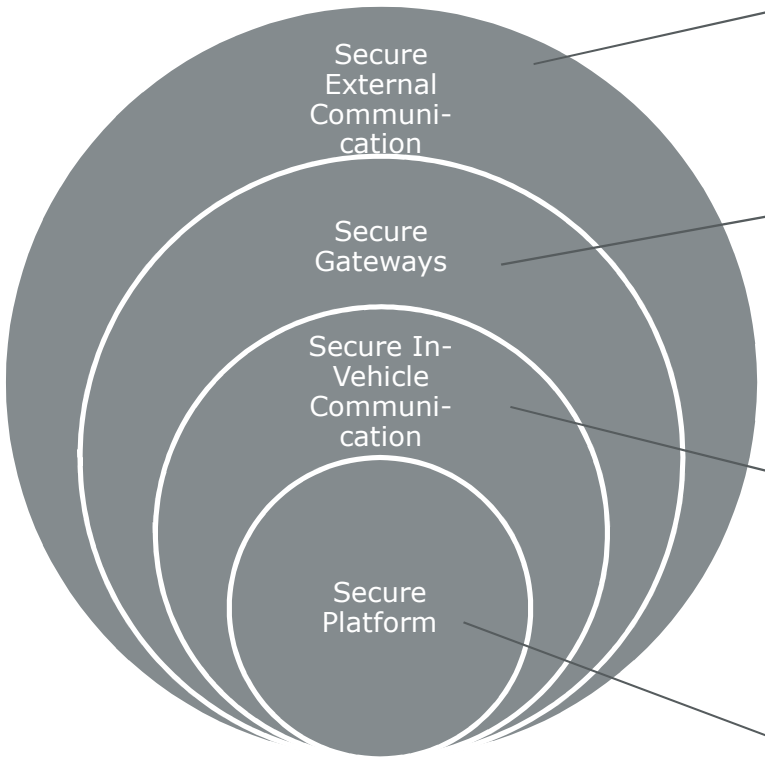
Security Level (SL)	Impact Level (IL)					
Threat Level (TL)		0	1	2	3	4
	0	QM	QM	QM	QM	Low
	1	QM	Low	Low	Low	Medium
	2	QM	Low	Medium	Medium	High
	3	QM	Low	Medium	High	High
	4	Low	Medium	High	High	Critical

Agenda

1. Introduction
2. Threat Analysis and Risk Assessment
3. Design for Security
4. Security Testing
5. Summary

Layered Security Concept

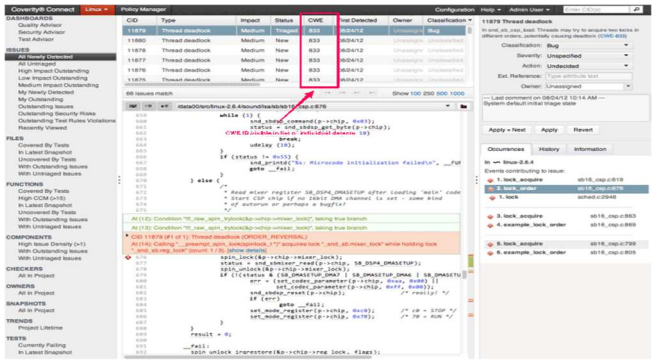
Legacy: Outside in



New development: Inside out

- Secure communication to services outside the vehicle
- Intrusion detection mechanisms
- Firewalls
- Key Infrastructure / Vehicle PKI
- Synchronized secure time
- Authenticity of messages
- Integrity and freshness of messages
- Confidentiality of messages
- Key storage
- Secure boot and secure update
- Crypto library
- HW trust anchor (HTA)

Security by Design: Secure Coding



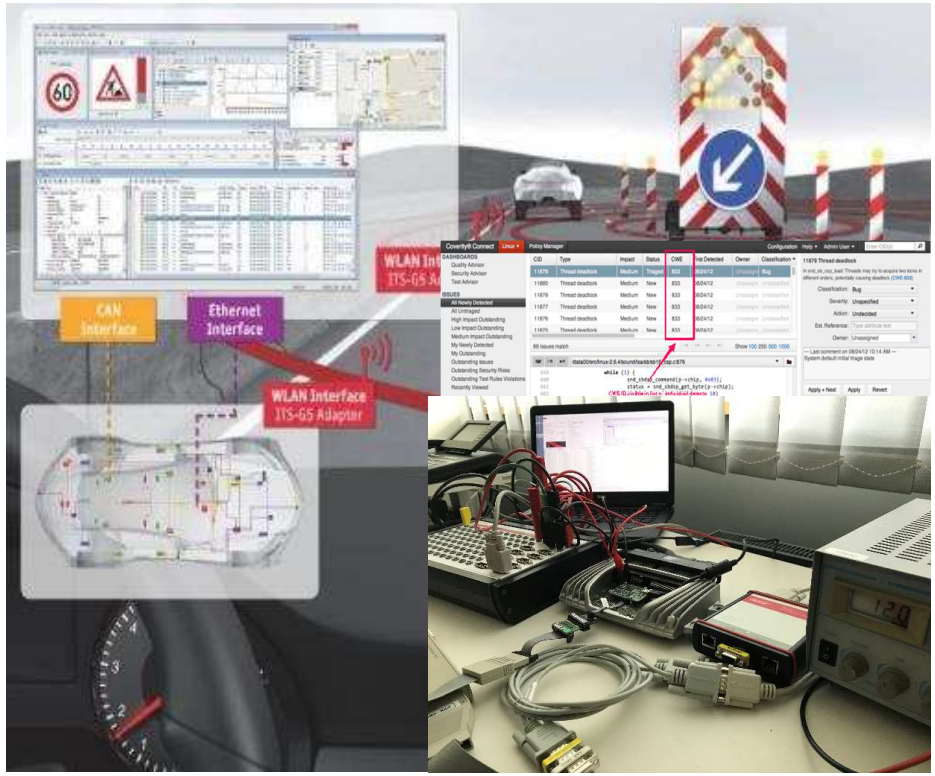
- **Goal**
  - **Avoid design and code errors which can lead to security exploits**

- Approach
  - **Use a hardened OS with secure partitioning**
    Avoid embedded Linux due to its complexity and rapid change and thus many security gaps, (e.g. NULL function pointer dereferences, which allow hackers to inject executable code).
  - **Deploy secure boot strategy**
    Starting with first-stage ROM loader with a pre-burned cryptographic key, the next levels are verified before executing to ensure authenticity of each component of the boot
  - **Apply rigorous static code analysis**
    Tools like Coverity, Klocwork or Bauhaus allow security checks, such as NULL pointer dereferences, memory access beyond allocated area, reads of uninitialized objects, buffer and array underflows, resource leaks etc.
  - **Use modified condition/decision coverage (MC/DC)**
    Detect backdoors

Security Implementation, Verification and Validation

- **Design**
  - Defensive coding, e.g. memory allocation, avoid injectable code, least privileges
  - Programming rules such as MISRA-C, SEI CERT
  - Trusted cryptographic algorithms
  - Key management and HW-based vaults for secrets
  - Awareness and governance towards social engineering

- **V&V Methods and Tools**
  - Static / dynamic code analyzer
  - Unit test with focused coverage, e.g. MC/DC
  - Interface scanner, layered fuzzing tester, encryption cracker, vulnerability scanner
  - Risk-based penetration testing



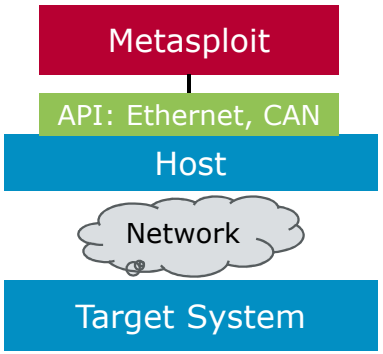
Classic structural coverage test is not sufficient. Test for the known – and for the unknown.

Agenda

1. Introduction
2. Threat Analysis and Risk Assessment
3. Design for Security
4. Security Testing
5. Summary

Security Validation: Penetration Testing Approach

1. Overview:
   - **Penetration Testing** is an offensive approach for security
   - Highly automated tools because a **high and growing** number of potential threads has to be systematically validated.
   - **Example: Metasploit** (*Open Source Framework*)



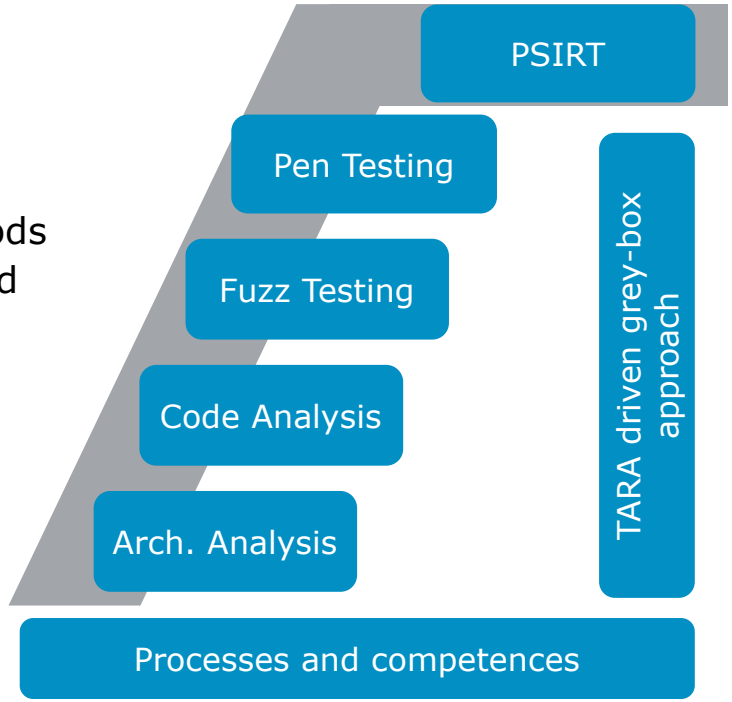
2. Basic Approach:
   - Scan the target system concerning vulnerabilities.
   - Select one of the proposed exploits, which make the weakness applicable.
   - Select and apply a payload (e.g. *meterpreter backdoor*) to get access to target ressources.

Permission of the target owner makes the difference between *penetration testing* and *hacking*.

VECTOR

Security by Lifecycle: Verification, Validation and Life-Cycle Management

- **PSIRT Collaboration (Product Security Incident Response Team)**
  - Handover, task assignments and distribution
- **OTA Updates: Ensure that each deployment satisfies security requirements**
  - Data encryption: Protection of intellectual property by encryption
  - Authorization: Protection against unauthorized ECU access
  - Validation: Safeguarding of data integrity e.g. in the flash memory
  - Authentication: Verification of authenticity through signature methods
  - Governance: Safety/security documentation is continuously updated
- **Pen Testing**
  - Connect with misuse, abuse and confuse cases
  - Vector Grey-Box PenTest based on TARA and risks
  - DoS, Replay, Mutant/Generated Messages
- **Fuzz Testing**
  - Brute-force CAN Fuzzer for fuzzing the Application SW
- **Analyses**
  - Code level: CQA, Coverage (e.g., VectorCAST)
  - Architecture level: Attack analysis, vulnerability analysis



Agenda

1. Introduction
2. Threat Analysis and Risk Assessment
3. Design for Security
4. Security Testing
5. Summary

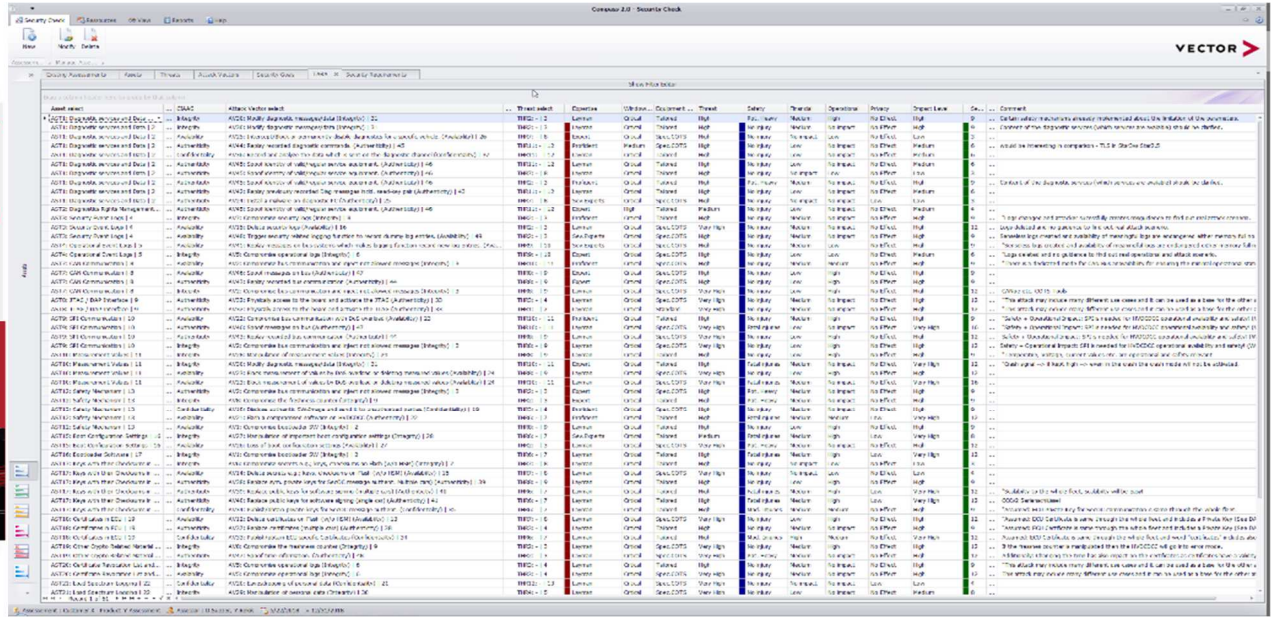
Summary

- **Security Context**
  - **ACES** (Autonomy, Connectivity, Efficiency, Services) depend of **effective cybersecurity**
  - **Standardization** of ISO/SAE 21434 is **in process**
  - Cybersecurity has to be implemented in **co-existence with functional safety**
- **Security Engineering**
  - **Cyber security goals** are derived by the Threat Analysis and Risk Assessment (**TARA**)
  - Cyber security embodies **layered scopes** from the secrets of the HTA to the public infrastructure
  - The security concepts implements **security mechanisms** to ascertain cybersecurity
- **Assurance of Security**
  - Architecture level **attack analysis**, stringent **coding guidelines** and gapless **code analysis**
  - Complementary set of test methods including **fuzz testing** and **penetration testing**
  - Comprehensive **security management** according governance regulations

Automotive cybersecurity remains is an evolving challenge

COMPASS: Vector Product for Security Check, TARA and Continuous Documentation

COMPASS information: www.vector.com/compass



Vector SecurityCheck facilitates

- **Systematic risk assessment and mitigation**
- **Traceability and Governance** with auditable risk and measure list
- **Heuristic checklists** with continuously updated threats and mitigation

Thank you for your attention.
For more information please contact us.

Passion. Partner. Value.

Vector Consulting Services

@VectorVCS
www.vector.com/consulting
consulting-info@vector.com
Phone: +49-711-80670-1520