



TASKING®

ISO 21434 - SECURE AUTOMOTIVE SOFTWARE DEVELOPMENT FROM A (COMPILER) TOOL PERSPECTIVE

Gerard Vink <gerard.vink@tasking.com>

TASKING <www.tasking.com>

Amersfoort

Oct 2021

Agenda

- 1 Importance & Urgence of this topic
- 2 ISO 21434 "Tool Qualification Requirements"
- 3 Processes used to identify compiler related cybersecurity risks
- 4 Examples of compiler related cybersecurity risks & mitigations



- Determined by **legislation** regarding type approval
 - In 2020 the United Nations Economic Commission for Europe (UNECE) released a **regulation** (not legally binding) on uniform provisions concerning the approval of vehicles with regards to cybersecurity and cybersecurity management system (aka WP.29)
 - In the European Union, the new regulation on cybersecurity will become **mandatory** (turned into legislation) for all **new** vehicle **types** from July **2022** and will become mandatory for all new vehicles **produced** from July **2024**
 - Cybersecurity will be **nonnegotiable for securing market access** and type approval in the future (product **liability** is of secondary importance)
- Regulation will affect the production of over 20 million cars yearly (excl. vans, trucks and busses)

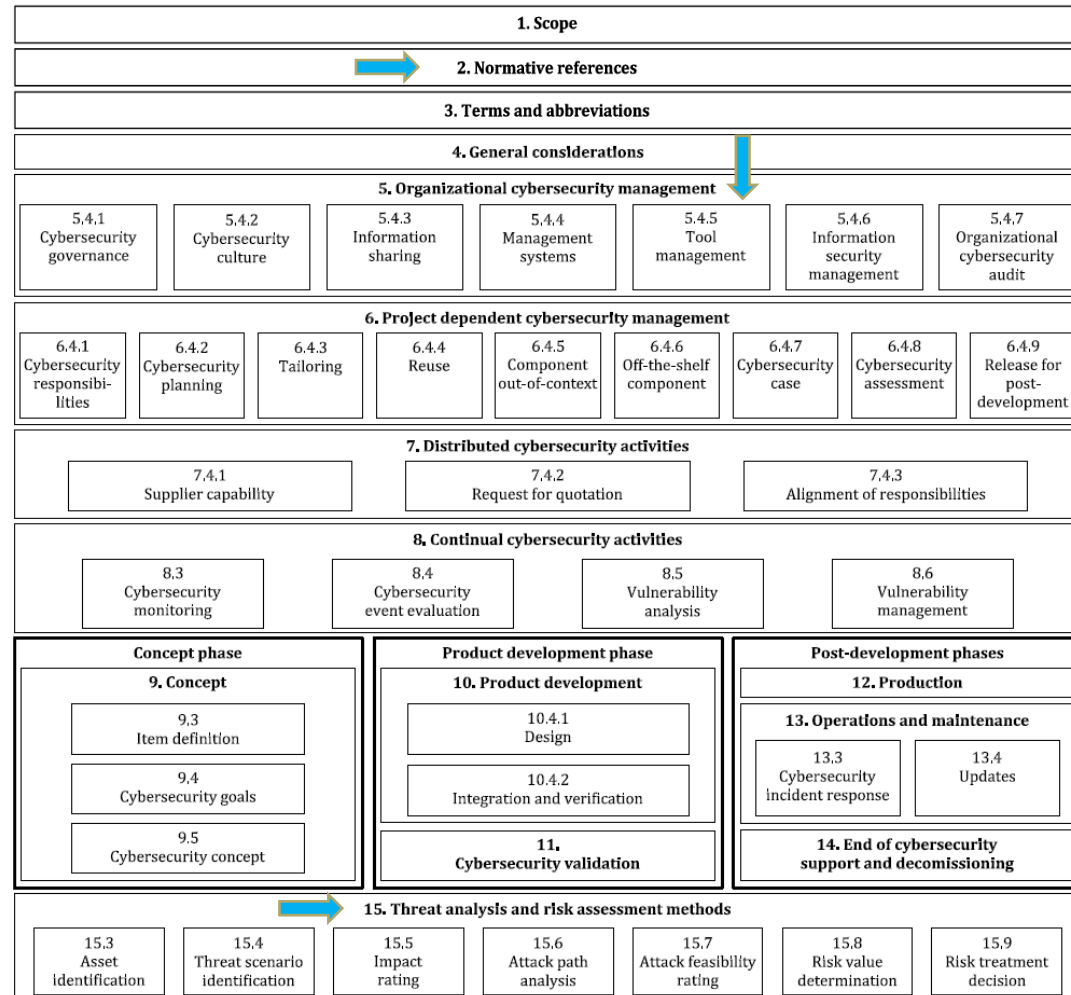
Importance & Urgence of Cybersecurity Compliance (cont.)

World Forum for Harmonization of Vehicle Regulations (WP.29) under the UN Economic Commission for Europe (UNECE)

■ Countries party to the 1958 agreement¹ (as of December 2018)



- **Japan & Korea** are members of the UNECE and are implementing the “cybersecurity regulation” into legislation
- The **USA** adopted the ISO/SAE 21434 Road vehicles — Cybersecurity Engineering standard, which was developed in cooperation between the ISO and the US based Society of Automotive Engineers (SAE) it replaces the Cybersecurity Guidebook for Cyber-Physical Vehicle Systems that was released in 2016.
- **China** is expected to follow



- Relevant in the context of this presentation are:
 - Chapter 2 **Normative** references
 - The cybersecurity standard contains normative references to parts of the ISO 26262 FuSa standard thereby **integrating** the **functional safety** (ISO 26262) and the **cybersecurity** (ISO 21434) lifecycles
 - Section 5.4.5 **Tool Management**
 - Tool qualification requirements can be deduced from this section. Details follow on next slide
 - Chapter 15 **Threat Analysis & Risk assessment** methods (TARA)
 - Guidance from this section shall be used to analyze cybersecurity related threats, risks, and remediations related to **Your software**
 - The **compilation process may influence the cybersecurity of your software**
 - **TASKING will do the analysis of compiler related cybersecurity threats, risks and remediations for you**



- Literal text from ISO 21434:5.4.5 Tool Management
 - 5.4.5 Tool Management
 - [RQ-05-14] **Tools** that can influence the cybersecurity of an item or component **shall be managed**.
 - NOTE: Such management can be established by:
 - Application of the user manual with errata;
 - **Protection against unintended usage or action**;
 - Access control for the tool users; and/or
 - Authentication of the tool
 - [RC-05-15] An appropriate environment to support remedial actions for cybersecurity incidents (see 13.3) should be reproducible until the end of cybersecurity support for the product.
- The tool, the compiler, shall be managed!
 - This is the **only** requirement that needs to be satisfied
 - How to interpret this requirement, **what are we supposed to do?**

To succeed in achieving or producing something despite difficult circumstances



- The ISO cybersecurity std does not describe a tool qualification process, therefore I (TASKING) do not know how to manage a tool
 - To solve this issue, I look to ISO 26262 which is frequently referenced by ISO 21434. ISO 26262 specifies several tool qualification methods:
 - Increased confidence from use
 - Evaluation of the development process
 - Tool validation
 - Development in accordance with a safety std.
 - For higher ASILs either method 3 or 4 shall be used
- So, **to qualify a tool for cybersecurity** related software development it shall be shown that either **the tool meets its cybersecurity related requirements** (tool validation) or that the tool is developed in accordance with a cybersecurity standard

- The process **applied by TASKING** is based on the Systems Security Engineering (SSE) Techniques published by the the National Cybersecurity FFRDC (NCF) part of MITRE, a non-profit organization funded by the US government
 - **FMEA**: to analyze the potential cybersecurity related risks that the TASKING compiler toolset can **introduce into the user's software**
 - The behavior of the compiled software shall not violate the **intentions of the user** under both "normal" conditions and under "cybersecurity attack" conditions
 - Analysis is done by engineers with an **in-depth knowledge** about the TASKING **compiler** and **its internals**
 - **TARA**: Threat and **Remediation** Analysis
 - The **CAPEC**, **CWE** and **CVE** databases provide input for this analysis. These databases contain known patterns of attack, known weakness types, and known cybersecurity vulnerabilities, and provide ways for risk assessment and remediation
- **TASKING** executes this cybersecurity analysis, **You** use the results

- The **output** of the FMEA and TARA process are:
 - Security related **requirements** to be **implemented** by the **supplier** of the toolset, and
 - Security related requirements (aka **guidelines**) to be **implemented** by the **user** of the toolset
- The **results** of the **compiler security qualification** are described in the Toolset's **Safety & Security** Manual and addresses a.o.:
 - The correct usage of the tool
 - By providing **guidelines** that explain how to **prevent or mitigate security related risks** associated with the compilation process and by **explaining** the **facilities** embedded in the compiler that **protect against cybersecurity attacks**
 - Protection against unintended usage or action
 - By implementing **facilities in the compiler** to protect against unintended actions and explaining the **residual risks** related to the compilation and optimization process

1. Risks & remediations related to coding style

- Many cybersecurity related risks can be **mitigated** by complying with **MISRA, CERT**, and/or AUTOSAR coding guidelines
 - When you develop automotive software, you already do comply with these guidelines
 - TASKING compilers support checks against MISRA and CERT rule sets
- Various **C Library** functions are **inherently unsafe & unsecure** due to their **specification**, their use shall be avoided
 - ISO-C11 and later Annex K provides alternative library functions that promote safer, more secure programming
 - TASKING will release a C library that is certified for the use in FuSa and cybersecurity related systems (ISO 26262 and ISO 21434 complaint)

2. Risks & remediations related to compiler optimizations

- Cybersecurity related risks can be introduced by compiler optimizations; particularly by exploiting **undefined behaviors**
 - Some examples:
 - Removal of code that does not affect the observable state of the program
 - Risk: sensitive data that is stored in memory is not cleared
 - Removal of code that causes undefined behavior; The following code can be *optimized(?)* away: ~~`if ( x > x+1 ) then { handle overflow }`~~
 - Risk: code that is supposed to detect integer arithmetic overflow is removed
 - Exploit UB of previously executed statements `int x=s->f; if(!s) return ERROR`
 - Risk: silently a discard null pointer check
 - For more examples see blogs [Undefined Behaviors in ISO-C: Their Effect on Your Embedded Software Part 1](#) and [Part 2](#)
- Remediation: compiler issues a diagnostic message to **inform the user** if an optimization affects the “intend of the programmer”

3. Remediations provided by specific compiler functionality

- Support the **security features** of the **target hardware**
 - **Secure boot** - Boot mode header support
 - Programming support for **Hardware Security Module**
- Facilitate **memory layout randomization** by generating position independent code & data (PIC/PID)
 - Note that the use of PIC/PID also may introduce cybersecurity risks
- Provide **stack canary** functionality to protect against stack smashing attacks
 - The use of stack canaries is not relevant for TriCore due to the separation of stack and context save area, but is relevant for the Parallel Processing Unit (PPU) that is introduced in the next generation of AURIX devices

- ISO 21434 compliance is urgent and needed to secure market access
- The compiler affects the cybersecurity of your system therefore “Tool management” is required



**IMPORTANCE
& URGENCY**



TASKING EXECUTES SECURITY ANALYSIS

- ISO 21434 compliance is urgent and needed to secure market access
- The compiler affects the cybersecurity of your system therefore “Tool management” is required
- A Threat analysis and Risk assessment shall be performed
- TASKING performs this TARA for you



**SAVES YOU WORK,
SHARED RESPONSIBILITY**

- ISO 21434 compliance is urgent and needed to secure market access
- The compiler affects the cybersecurity of your system therefore “Tool management” is required
- A Threat analysis and Risk assessment shall be performed
- TASKING performs this TARA for you



**FOLLOW THE
GUIDELINES**

- ISO 21434 compliance is urgent and needed to secure market access
- The compiler affects the cybersecurity of your system therefore “Tool management” is required
- A Threat analysis and Risk assessment shall be performed
- TASKING performs this TARA for you
- You follow the guidelines provided in the TASKING “Safety & Cybersecurity manual”



No need to acquire specialized competences and knowledge
Increased quality of the cybersecurity analysis
Cost and lead-time reduction

Thank You
Questions: gerard.vink@tasking.com